



Programa de Pós-Graduação em

Computação Aplicada

Mestrado/Doutorado Acadêmico

Gefersom Cardoso Lima

Segmentação de fácies sísmicas com redes neurais

São Leopoldo, 2021

L732s Lima, Gefersom Cardoso.
Segmentação de fácies sísmicas com redes neurais /
Gefersom Cardoso Lima. – 2021.
134 f. : il. ; 30 cm.

Dissertação (mestrado) – Universidade do Vale do Rio
dos Sinos, Programa de Pós-Graduação em Computação
Aplicada, 2021.

“Orientador: Prof. Dr. Gabriel de Oliveira Ramos
Coorientador: Prof. Dr. Sandro José Rigo.”

1. Sísmica. 2. Interpretação. 3. Segmentação. 4. Rede
neural. 5. Codificador-decodificador. I. Título.

CDU 004.732

Dados Internacionais de Catalogação na Publicação (CIP)
(Bibliotecária: Amanda Schuster – CRB 10/2517)

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior Brasil (CAPES) - Código de Financiamento 001 /"This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001

ATA DE BANCA EXAMINADORA DE DISSERTAÇÃO DE MESTRADO Nº 07/2021

Aluno: Gefersom Cardoso Lima

Título da Dissertação: “SEGMENTAÇÃO DE FÁCIES SÍSMICAS COM REDES NEURAIIS”

Banca: Prof. Dr. Gabriel de Oliveira Ramos – Orientador
Prof. Dr. Sandro José Rigo - Coorientador
Prof. Dr. Gustavo Pessin - UFOP
Prof. Dr. Rafael Kunst - UNISINOS

Aos sete dias do mês de abril do ano de 2021, às 9h, a Comissão Examinadora de Defesa de Dissertação composta pelos professores: Prof. Dr. Gabriel de Oliveira Ramos, Orientador – UNISINOS (participação por webconferência); Prof. Dr. Sandro José Rigo, Coorientador – UNISINOS (participação por webconferência); Prof. Dr. Gustavo Pessin, Membro da Banca – UFOP (participação por webconferência) e Prof. Dr. Rafael Kunst, Membro da Banca – UNISINOS (participação por webconferência), para analisar e avaliar a Dissertação apresentada pelo Gefersom Cardoso Lima (participação por webconferência).

Considerações da Banca:

Após deliberação, a banca reconheceu a qualidade do trabalho bem como sua adequação para uma dissertação de mestrado. Foram sugeridas diversas melhorias que poderão ser consideradas para a versão final do texto.

Ocorreu alteração do título? (X)Não () Sim

Indicar o novo título:

A Banca Examinadora, em cumprimento ao requisito exigido para a obtenção do Título de Mestre em Computação Aplicada, julga esta dissertação:

(X) APROVADA () REPROVADA

Conforme Artigo 67 do Regimento do Programa o texto definitivo, com aprovação do Orientador, deverá ser entregue no prazo máximo de sessenta (60) dias após a defesa. O resultado da banca é de consenso entre os avaliadores. A emissão do Diploma está condicionada a entrega da versão final da Dissertação. A sessão da Defesa de Dissertação ocorreu integralmente por webconferência para atender às recomendações da OMS e Ministério da Saúde com relação ao Covid-19.

São Leopoldo, 07 de abril de 2021



Prof. Dr. Gabriel de Oliveira Ramos - Orientador

Aos nossos pais.

*If I have seen farther than others,
it is because I stood on the shoulders of giants.*
— SIR ISAAC NEWTON

AGRADECIMENTOS

Primeiramente agradece a Deus, pois se não fosse da vontade Dele, hoje esse trabalho não estaria concluído.

Aos meus pais pelos inúmeros esforços para que eu pudesse alcançar os meus objetivos sempre me apoiando e servindo de inspiração.

Ao meu orientador Gabriel de Oliveira Ramos por ter aceitado me orientar e pelas inúmeras contribuições neste trabalho e nos artigos submetidos. Ao meu coorientador Sandro José Rigo pelo apoio e guiamento durante a fase inicial do curso e por ter me indicado ao Gabriel para ser seu orientando.

A CAPES pelo apoio financeiro no desenvolvimento deste trabalho, apoio esse que foi de fundamental importância para possibilitar a realização do curso.

“You cannot open a book without learning something”.
(Confucius)

RESUMO

A interpretação de dados sísmicos é importante para a caracterização da forma dos sedimentos de uma área de estudo geológico. Tradicionalmente, esse trabalho é realizado escolhendo-se visualmente pontos que representem os limites de fácies sísmicas e executando-se uma ferramenta para realizar a inferência dos outros pontos limites. Esse processo requer uma grande quantidade de trabalho manual e pode permitir que algumas fácies não sejam identificadas, tornando o trabalho resultante menos detalhado do que poderia ser. Com o aumento do uso de aprendizagem profunda focada na segmentação de imagens, a sua aplicação no auxílio à interpretação sísmica pode trazer ganhos ao diminuir o trabalho manual e o tempo gasto ao se realizar o estudo de uma área geológica. Assim, nesse trabalho fizemos um estudo da aplicação de redes neurais profundas do tipo codificador-decodificar para a identificação das linhas separatórias de sismofácies. Como resultado, criamos uma rede neural chamada DNFS, a qual é baseada na U-Net e StNet, possui menos parâmetros que estas e é voltada a segmentação binária de dados sísmicos. Esse tipo de segmentação nos permitiu segmentar um número arbitrário de fácies sísmicas somente se focando na transição entre elas. Para usar a segmentação binária utilizamos um método simples de adaptação dos conjuntos de dados sobre os quais fizemos os experimentos. Essa adaptação utiliza linhas pretas entre as interseções das fácies sísmicas e cor branca para todo o restante da imagem rotulada. Para o cálculo de perda utilizamos uma função composta pela combinação linear das funções *cross-entropy* e *Jaccard loss*. Para otimizar o coeficiente da combinação linear da função que pondera o peso de *cross-entropy* e *Jaccard loss* no valor de perda, realizamos vários experimentos tendo como resultado que se a *cross-entropy* contribuir com 75% e *Jaccard loss* com 25%, poderíamos obter predições com alta fidelidade das linhas separatórias entre as fácies sísmicas. Também realizamos uma extensa avaliação experimental e ajustes dos hiperparâmetros e comparamos os resultados com as redes bases U-Net e StNet aplicadas sobre os mesmos conjuntos de dados. Ao final, obtivemos uma rede neural que pode se treinada em aproximadamente 15 minutos e oferece um índice acima de 95% relativo à métrica IoU sobre os conjuntos de dados *StData-12* e *Facies-Mark*.

Palavras-chave: sísmica, interpretação, segmentação, rede neural, codificador-decodificador.

ABSTRACT

The interpretation of seismic data is important for the characterization of the shape of the sediments in a geological study area. Traditionally, this work is carried out by visually choosing points that represent the limits of seismic facies and executing a tool to make the inference of other limit points. This process requires a lot of manual labor and can allow some facies to go unidentified, making the resulting work less detailed than it could be. With the increase in the use of deep learning focused on image segmentation, its application in helping seismic interpretation can bring gains by decreasing manual work and the time spent when studying a geological area. Thus, in this work we made a study of the application of deep neural networks of the encoder-decode type for the identification of seismic facies separating lines. As a result, we created a neural network called DNFS, which is based on U-Net and StNet, has fewer parameters than these and is aimed at binary segmentation of seismic data. This type of segmentation allowed us to segment an arbitrary number of seismic facies just by focusing on the transition between them. To use binary segmentation we use a simple method of adapting the data sets on which we did the experiments. This adaptation uses black lines between the intersections of the seismic facies and white color for the rest of the labeled image. For the calculation of loss we use a function composed by the linear combination of the cross-entropy and Jaccard loss functions. To optimize the coefficient of the linear combination of the function that weighs the weight of cross-entropy and Jaccard loss in the loss value, we performed several experiments with the result that if the cross-entropy contributes 75 % and Jaccard loss with 25 %, we could obtain predictions with high fidelity of the separating lines between the seismic facies. We also carried out an extensive experimental evaluation and adjustments of the hyper-parameters and compared the results with the base networks U-Net and StNet applied on the same data sets. In the end, we obtained a neural network that can be trained in approximately 15 minutes and offers an index above 95% relative to the IoU metric on the StData-12 and Facies-Mark datasets.

Keywords: seismic, interpretation, segmentation, neural network, encoder-decoder.

LISTA DE FIGURAS

1	Comparação de dados médicos e seções sísmicas. Em ambos os casos, as linhas são usadas para delimitar as regiões de interesse (por exemplo, células e fácies sísmicas).	32
2	Geração de ondas sísmicas.	38
3	Representação de uma interface entre dois meios diferentes. Impedância acústica: Z_1 e Z_2 . Densidade: p_1, p_2 . Velocidade de propagação de onda: V_1, V_2	38
4	Leiaute de aquisição em terra. Rn: número do receptor; em (b) traços sísmicos simplificados correspondentes as interfaces em (a). Adaptado de (AYAN; YADAV, 2018)	39
5	Esquema de aquisição sísmica em terra.	40
6	Aquisição sísmica no mar.	40
7	Exemplo de interpretação estrutural. As linhas coloridas representam as geometrias identificadas na seção sísmica.	41
8	Exemplo de um volume sísmico.	42
9	Exemplo de algumas seções sísmicas que compõem o volume da Figura 8.	42
10	Modelo simplificado de neurônio biológico.	43
11	Exemplo de perceptron.	44
12	Modelo de rede neural.	44
13	Mapas de propriedades criados a partir da aplicação de quatro filtros.	49
14	Aplicação de convolução sobre uma imagem 2D.	50
15	Aplicação de <i>max-pooling</i> sobre um mapa de propriedades.	51
16	Exemplo da aplicação da técnica de <i>dropout</i>	51
17	Exemplo de rede com camada totalmente conectada.	52
18	Processo geral de um <i>encoder-decoder</i>	53
19	Processo geral de um <i>encoder-decoder</i>	54
20	Segmentação de três tipos de objetos: grama, cachorro e ovelha. Em verde está segmentada a ovelha, em vermelho o cachorro, em preto a grama.	55
21	<i>Grid</i> representativo de uma imagem em escala de cinza. À esquerda os valores da escala de cinza compoendo o <i>grid</i> , à direita a imagem criada a partir do <i>grid</i>	56
22	Exemplo de formato para criação de <i>patches</i> que podem ser utilizado para <i>patch-classification</i> ou <i>pixel-to-pixel classification</i> . As linhas vermelhas delimitam a áreas dos <i>patches</i> . Na horizontal estão representadas as colunas e na vertical as linhas. A cor branca representa a área onde há um domo de sal, já a cor preta a área onde não há. a) Imagem sísmica e a representações de <i>patches</i> quadrados; b) máscara da imagem sísmica contendo um domo de sal; c) Colunas 1 e 3 representam os <i>patches</i> das imagens sísmica e as colunas 2 e 4, as máscaras correspondentes.	57
23	Exemplo de rotulação de <i>pixels</i> de um <i>patch</i> . À esquerda se encontra o <i>patch</i> . À direita a rotulação do <i>patch</i> onde em cada posição onde há a cor preta foi inserido o rótulo 1 (um) e nas de cor branca o rótulo 0 (zero).	58
24	Estágios do protocolo de revisão.	59
25	Palavras-chave para a realização de buscas em bases de dados.	61
26	Processo de pesquisa e seleção de artigos	67
27	Quantidade de trabalhos publicados por ano.	69
28	Esquema da variação de parâmetros da rede DNFS	88

29	Estrutura da rede StNet.	88
30	Estrutura da rede U-Net.	89
31	Imagens do <i>dataset StData</i> e a adaptação das máscaras feita neste trabalho. À direita se encontra a imagem da seção sísmica de entrada, no meio a imagem rotulada de uma seção sísmica utilizada pelo trabalho de Di, Gao e AlRegib (2019a). A imagem mais à direita é a adaptada para o uso neste trabalho, a qual passou a ser constituída da cor branca para o fundo e de linhas separatórias entre as áreas que antes eram coloridas.	92
32	Exemplos de segmentos de imagem sísmica e máscara do <i>dataset StData-12</i> . As colunas ímpares correspondem aos segmentos adaptados a partir das seções sísmicas; as colunas pares correspondem aos segmentos das máscaras criadas para cada imagem adjacente à esquerda.	93
33	Seções sísmicas do <i>dataset StData-12</i> . À direita se encontra a imagem da seção sísmica de presente no <i>dataset StData-12</i> . A imagem mais à direita é a adaptada para o uso neste trabalho, a qual passou a ser constituída da cor branca para o fundo e de linhas separatórias entre as áreas que antes eram coloridas.	94
34	À direita se encontra a imagem da seção sísmica de entrada, no meio a imagem rotulada de uma seção sísmica do <i>dataset Facies-Mark</i> . A imagem mais à direita é a adaptada para o uso neste trabalho, a qual passou a ser constituída da cor branca para o fundo e de linhas separatórias entre as áreas que antes eram coloridas.	95
35	Exemplos de segmentos de imagem sísmica e máscara do <i>dataset Facies-Mark</i> . As colunas ímpares correspondem aos segmentos adaptados a partir das seções sísmicas; as colunas pares correspondem aos segmentos das máscaras criadas para cada imagem adjacente à esquerda.	95
36	Comparação das previsões feitas pelas redes DNFS (com fator 8), StNet e U-Net com diferentes valores do coeficiente ψ para a função de perda composta (Eq.(4.5)).	103
37	Curvas de aprendizagem para DNFS com fator 4, 8, 16 e 32. Evolução do índice <i>Jaccard</i> (eixo vertical) durante o treinamento das redes DNFS, StNet e U-Net com a progressão dos <i>epochs</i> (eixo horizontal).	104
38	Curvas de aprendizagem para DNFS com fator 64, StNet e U-Net. Evolução do índice <i>Jaccard</i> (eixo vertical) durante o treinamento das redes DNFS, StNet e U-Net com a progressão dos <i>epochs</i> (eixo horizontal).	105
39	Esquema de variação de mapas de propriedades na StNet e o uso de <i>skip-connections</i> . As barras azuis representam a quantidade de filtros nos blocos que compõem a StNet. As setas de cor azul as transferências de propriedades entre camadas internas de blocos e as vermelhas as transferências entre blocos.	108
40	Predição do <i>inline 390</i> do <i>dataset StData-12</i> com a rede DNFS com fator 4.	109
41	Predição do <i>inline 390</i> do <i>dataset StData-12</i> com a rede DNFS com fator 8.	109
42	Predição do <i>inline 390</i> do <i>dataset StData-12</i> com a rede DNFS com fator 16.	110
43	Predição do <i>inline 390</i> do <i>dataset StData-12</i> com a rede DNFS com fator 32.	110
44	Predição do <i>inline 390</i> do <i>dataset StData-12</i> com a rede DNFS com fator 64.	111
45	Predição do <i>inline 390</i> do <i>dataset StData-12</i> com a rede StNet.	111
46	Predição do <i>inline 390</i> do <i>dataset StData-12</i> com a rede U-Net.	112
47	Predições de um <i>inline</i> presente no <i>dataset Facies-Mark</i> feitas pela DNFS com fator 4.	113

48	Predições de um <i>inline</i> presente no <i>dataset</i> Facies-Mark feitas pela DNFS com fator 8.	114
49	Predições de um <i>inline</i> presente no <i>dataset</i> Facies-Mark feitas pela DNFS com fator 16.	114
50	Predições de um <i>inline</i> presente no <i>dataset</i> Facies-Mark feitas pela DNFS com fator 32.	115
51	Predições de um <i>inline</i> presente no <i>dataset</i> Facies-Mark feitas pela DNFS com fator 64.	115
52	Predições de um <i>inline</i> presente no <i>dataset</i> Facies-Mark feitas pela rede StNet.	116
53	Predições de um <i>inline</i> presente no <i>dataset</i> Facies-Mark feitas pela rede U-Net.	116

LISTA DE TABELAS

1	Questões de pesquisa	60
2	Consultas feitas nas bases de dados e períodos de filtragem.	62
3	Critério de inclusão	63
4	Critério de exclusão	64
5	Critérios de qualidade.	65
6	Passos da seleção de trabalhos	66
7	Formulário de extração de dados	68
8	Resultados da avaliação de qualidade dos artigos selecionados.	80
9	Sumário das técnicas utilizadas nos trabalhos analisados, parte 1 de 3.	81
10	Sumário das técnicas utilizadas nos trabalhos analisados, parte 2 de 3.	82
11	Sumário das técnicas utilizadas nos trabalhos analisados, parte 3 de 3.	83
12	Métricas de avaliação utilizadas nos trabalhos. Primeiro campo se refere a métrica, o segundo campo, o identificador do trabalho, terceiro campo a quantidade de artigos que utilizaram a métrica. O item com * representa que não foi utilizada uma métrica de avaliação, em seu lugar o autor de trabalho fez uma análise visual dos resultados.	83
13	Datasets utilizados para treinamento das redes neurais	84
14	Características resumidas dos trabalhos analisados. Os artigos que compõem a tabela são aqueles que possuem alta correlação com este trabalho.	84
15	Parâmetros de treinamento e tamanhos de rede neural para DNFS, StNet e U-Net. O asterisco se refere aos resultados das arquiteturas StNet e U-Net propostas em RONNEBERGER; FISCHER; BROX (2015) e DI; GAO; ALREGIB (2019a).	100
16	Comparação de desempenho de DNFS, StNet e U-Net, em termos de tempo de treinamento, precisão e porcentagem de <i>pixels</i> pretos corretos. O melhor resultado da arquitetura DNFS (quando combinamos o tempo de treinamento e a porcentagem de <i>pixels</i> pretos) é destacado em negrito.	106
17	Variação do valor de <i>dropout</i> e índices de <i>pixels</i> pretos preditos corretamente. A célula da tabela com asterisco (*) representa a soma dos tempos de treinamento das redes. Em negrito está a melhor média de <i>pixels</i> pretos em relação ao <i>dropout</i> . Parte 1 de 3.	131
18	Variação do valor de <i>dropout</i> e índices de <i>pixels</i> pretos preditos corretamente. A célula da tabela com asterisco (*) representa a soma dos tempos de treinamento das redes. Parte 2 de 3.	132
19	Variação do valor de <i>dropout</i> e índices de <i>pixels</i> pretos preditos corretamente. A célula da tabela com asterisco (*) representa a soma dos tempos de treinamento das redes. Parte 3 de 3.	133
20	Variação do valor do <i>learning rate</i> e índices de <i>pixels</i> pretos preditos corretamente. A célula da tabela com asterisco (*) representa a soma dos tempos de treinamento das redes.	134

LISTA DE ABREVIATURAS

ANN	<i>Artificial neural networks</i>
API	<i>Application Programming Interface</i>
coord.	coordenador
CNN	<i>Convolution neural network</i>
CPU	<i>Central Process Unit</i>
CQ	Critério de qualidade
DNFS	Deep neural network for facies segmentation
ED	<i>Encoder-decoder</i>
Eq	Equação
FCNN	<i>Fully convolution neural network</i>
GAN	<i>Generative adversarial network</i>
Fig.	Figura
Geo	<i>Geology</i>
GPU	<i>Graphic processing unit</i>
IoU	<i>Intersection over Union</i>
LR	<i>Learning rate</i>
PC	<i>Patch-classification</i>
PPC	<i>Pixel-to-pixel classification</i>
QP	Questão de pesquisa
QA	Avaliação da qualidade
RSL	Revisão sistemática da literatura
RAM	<i>Random access memory</i>
SAE	<i>Sparcy auto-encoder</i>

LISTA DE SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
AAPG	<i>American Association of Petroleum Geologists</i>
arXiv	<i>Open-access repository of electronic preprints</i>
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
EAGE	<i>European Association of Geoscientists and Engineers</i>
FAPERGS	Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul
ICASSP	<i>International Conference on Acoustics, Speech and Signal Processing</i>
ICIP	<i>International Conference on Image Processing</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IJCNN	<i>International Joint Conference on Neural Networks</i>
ISPRS	<i>International Journal of Geo-Information</i>
SEG	<i>Society of Exploration Geophysicists</i>
SEG-Y	<i>The Society of Exploration Geophysicists - Y</i>
SPE	<i>Annual Technical Conference and Exhibition</i>

LISTA DE SÍMBOLOS

2D *Two dimensions*

3D *Three dimensions*

H Hora

MB *Mega Byte*

mm Minuto

s Segundo

SUMÁRIO

1 INTRODUÇÃO	29
1.1 Contexto e motivação	29
1.2 Questão de pesquisa	32
1.2.1 Objetivo geral	32
1.3 Objetivos específicos	33
1.4 Contribuições	33
1.5 Publicações	34
1.6 Estrutura do texto	34
2 FUNDAMENTAÇÃO TEÓRICA	37
2.1 Sísmica de reflexão	37
2.2 Redes neurais	43
2.2.1 Redes convolucionais	48
2.2.2 Arquitetura <i>encoder-decoder</i>	53
2.3 Segmentação com redes neurais	55
2.3.1 Abordagens de segmentação semântica com redes neurais	55
2.4 Considerações	57
3 TRABALHOS RELACIONADOS	59
3.1 Protocolo da revisão de literatura	59
3.2 Questões de pesquisa	60
3.3 Estratégias de busca	61
3.4 Seleção de trabalhos	63
3.4.1 Critérios de inclusão e exclusão	63
3.4.2 Critérios de qualidade	64
3.4.3 Resultados	65
3.5 Extração dos dados	65
3.6 Síntese dos dados	66
3.7 Resultados	67
3.7.1 Resultados da avaliação de qualidade	67
3.7.2 Informações gerais sobre os trabalhos	69
3.7.3 RQ1 - Abordagens de <i>Deep learning</i>	70
3.7.4 RQ2 - Algoritmos e arquiteturas utilizadas	72
3.7.5 RQ3 - Métricas	75
3.7.6 RQ4 - Representação de dados sísmicos	75
3.7.7 RQ5 - Comparação entre modelos	76
3.8 Considerações	78
4 MÉTODO	85
4.1 Arquitetura	86
4.2 Função de perda	87
4.3 Dados de treinamento	90
4.4 Treinamento	96
4.5 Considerações	97

5	AVALIAÇÃO EXPERIMENTAL E RESULTADOS	99
5.1	Metodologia	99
5.2	Otimização do coeficiente de perda	101
5.3	Resultados e discussão	106
5.4	Considerações	117
6	CONCLUSÕES E TRABALHOS FUTUROS	119
6.1	Limitações	120
6.2	Trabalhos futuros	120
	REFERÊNCIAS	123
	APÊNDICE A – INFORMAÇÕES COMPLEMENTARES	131
A.1	Valores de <i>dropout</i>	131
A.2	Valores de <i>learning rate</i>	134

1 INTRODUÇÃO

1.1 Contexto e motivação

O petróleo e o gás natural desempenham um papel importante na economia mundial, sendo empregados nos mais diversos produtos, desde alimentos até dispositivos eletrônicos. A exploração de hidrocarbonetos é a área responsável pela busca e exploração de depósitos de petróleo e gás natural. Uma tarefa crucial na exploração de hidrocarbonetos refere-se à interpretação sísmica, que visa identificar as diferentes características e estruturas sedimentares na exploração de petróleo e gás (ASHCROFT, 2011).

Em fluxos de trabalho convencionais, os dados sísmicos são interpretados manualmente por um especialista humano, que identifica o contato entre diferentes tipos de rocha e feições geológicas (LIU et al., 2020a). Este processo geralmente leva muito tempo para ser concluído (pode durar meses) devido à quantidade de dados a serem interpretados e depende muito do conhecimento e da experiência dos intérpretes (WALDELAND et al., 2018). Algumas ferramentas pretendem identificar padrões nos dados sísmicos para diminuir o tempo e reduzir o viés nas interpretações sísmicas. Essas ferramentas são baseadas em atributos sísmicos que destacam ou quantificam interesses de interpretação e permitem ao intérprete inferir o ambiente estrutural ou deposicional. Esses atributos visam capturar a experiência de reconhecimento de padrões do intérprete, quantificando a amplitude e características morfológicas vistas nos dados sísmicos (CHOPRA; MARFURT, 2007). Alguns exemplos desses atributos incluem coerência, que captura as mesmas descontinuidades vistas nos dados sísmicos e são interpretadas como falhas (GAO, 2013); fase instantânea (IP), que detecta descontinuidades estruturais em imagens sísmicas (YUAN et al., 2019); textura estratigráfica caótica, que identifica chaminés de gás (RANDEN; SØNNELAND, 2005); envelope, que detecta mudanças litológicas causadas por fortes reflexos de energia e limites de sequência (BABASAFARI; SAMADIAN, 2012). Uma desvantagem significativa dessas ferramentas é que elas não são criadas para melhorar automaticamente seus resultados e, portanto, dependem da interferência manual do intérprete. Em outras palavras, o intérprete pode entrar em um processo de tentativa e erro, onde o mesmo testa a aplicação de vários atributos sísmicos para verificar quais fornecem resultados mais adequados para a área estudada.

Na tentativa de amenizar esse problema, alguns trabalhos fizeram uso de redes neurais e segmentação de imagens. A segmentação é um processo que resulta na delimitação dos objetos contidos nas imagens (GONZALEZ; WOODS, 2018). A aplicação de redes neurais para segmentação em dados sísmicos fornece a capacidade de aprender as características essenciais que permitem a construção de uma imagem de saída contendo estruturas geológicas segmentadas. O uso de redes neurais convolucionais profundas (CNNs) tem se mostrado particularmente útil no contexto da segmentação de fácies sísmicas (MA et al., 2019; SHI; WU; FOMEL, 2018, 2019; WALDELAND et al., 2018; ALAUDAH et al., 2019). Em geral, esses trabalhos usam

um tipo de estrutura CNN chamada *encoder-decoder*, onde o *encoder* (codificador) extrai um mapa de características das imagens de entrada, que é então usado pelo *decoder* (decodificador) para construir uma imagem, podendo esta ser a de entrada. Camadas de convolução compõem o *encoder* e camadas de convolução transpostas compõem o *decoder* (WANI et al., 2019). Com esse tipo de rede neural, pode-se simular a forma de trabalho do intérprete em algum nível. O intérprete analisa imagens sísmicas para identificar características estratigráficas e estruturais, geometria e contatos litológicos (NANDA, 2016). Com isso em mente, poderíamos treinar uma rede neural *encoder-decoder* para identificar *pixels* entre geometrias de fácies sísmicas e extrair características essenciais para reconstruir uma saída com regiões segmentadas representando geometrias de fácies sísmicas separadas por linhas. A maioria dos trabalhos que usam redes neurais para essa tarefa enfoca a segmentação de apenas um tipo de fácies sísmica e usa uma função de perda que não considera a distribuição espacial dos *pixels* nas previsões de resultados.

Neste trabalho, apresentamos *Deep Neural Network for Facies Segmentation* (DNFS, para abreviar). DNFS é uma variante de U-Net (RONNEBERGER; FISCHER; BROX, 2015) e StNet (DI; GAO; ALREGIB, 2019a). A U-Net tem sido amplamente utilizada para segmentação de imagens e sua arquitetura foi originalmente proposta para a segmentação de imagens médicas. Embora os domínios médico e geológico possam parecer distantes um do outro, as imagens consideradas compartilham uma natureza semelhante. Como mostrado na Figura 1, tal similaridade pode ser descrita em termos de presença de ruído, texturas semelhantes, objetos separáveis por linhas e escalas de cores semelhantes. A StNet, foi desenvolvida para a segmentação de fácies sísmicas com base na similaridade de textura. A principal diferença entre DNFS e UNet refere-se aos seus hiper-parâmetros e às camadas convolucionais transpostas na parte do *decoder* da rede. Em comparação com a StNet, a DNFS usa mais blocos de convolução e convolução transposta com menos filtros e conexões de ligação entre as partes *encoder-decoder*. Além disso, essa conexão é composta apenas por uma camada convolucional.

Junto a DNFS propomos que a simples adaptação de interpretações sísmicas contidas em um *dataset* através do uso de linhas pretas como divisórias entre as interfaces das fácies sísmicas sobre um fundo branco poderia otimizar a quantidade de parâmetros a serem treinados e impactar na redução do tempo de treinamento da rede. Com isso, poderíamos ter um rede enxuta na quantidade de parâmetros e que poderia ser treinada em pouco. Isso nos possibilitaria a inserção da rede no fluxo diário de trabalho de intérpretes de dados sísmicos, onde poderiam realizar várias interpretações e utilizar um rede neural treinada em poucos minutos para expandir as interpretações realizadas às outras áreas do volume sísmico que não foram analisadas. O intérprete poderia também corrigir as já feitas e realizar diferentes interpretações sobre os mesmos dados sísmicos. Para isso, fizemos vários experimentos mudando pragmaticamente a quantidade de parâmetros da DNFS e analisamos o tempo de treinamento e os valores de índices IoU fornecidos pelas variações da DNFS. A partir desses experimentos, chegamos à configurações da DNFS que podem ser treinadas em menos de 30 minutos e que são capazes de fornecer as predições das linhas divisórias entre as fácies sísmicas. Os nossos testes foram

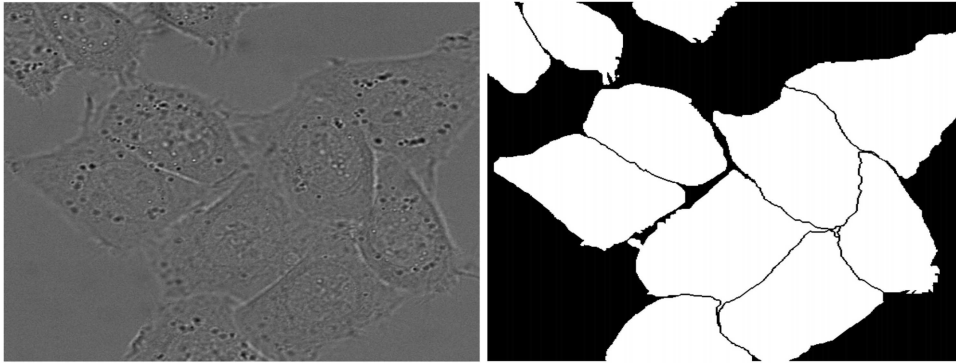
baseados nos *datasets StData-12* (DI; GAO; ALREGIB, 2019b) e *Facies-Mark* (ALFARRAJ, 2019) os quais foram adaptados para a utilização de segmentação binária através das linhas pretas em fundo branco. Nos concentramos na otimização do tempo de treinamento e na quantidade de parâmetros da rede devido à característica de que uma rede treinada para uma determinada área não é capaz de generalizar as suas predições para outra área pois, mesmo que as geometrias das fácies sísmicas sejam semelhantes, os valores de impedância acústica que as geraram podem ser diferentes da área utilizada para treinar a rede. Estes valores dependem da composição dos materiais existentes nas subsuperfícies da área estudada, o que impacta nos valores de impedância captados durante a aquisição dos dados sísmicos.

Como observamos ao longo da realização da revisão bibliográfica, a maioria dos trabalhos utiliza uma função de perda inadequada para se realizar a segmentação de objetos em dados sísmicos. Essa função de perda foi a *cross-entropy* (BOER et al., 2005) e esta tem como objetivo associar classes a *pixels* e desconsidera a distribuição espacial dos mesmos. Assim, realizamos a combinação linear de *cross-entropy* e *Jaccard loss* (RAHMAN; WANG, 2016) neste trabalho. Onde a *Jaccard loss* é responsável por verificar a distribuição espacial dos *pixels* preditos pela rede. Para encontrar qual peso a ser dado à *cross-entropy* e a *Jaccard loss* realizamos experimentos definindo diferentes pesos para cada uma dessas funções e escolhemos a configuração que forneceu melhores índices de *pixels* pretos preditos corretamente. Esse índice foi obtido através da criação de uma nova métrica de avaliação utilizada após as predições das redes terem sido feitas. Essa métrica considerou somente os *pixels* pretos preditos nas posições corretas na máscara de saída. Ou seja, os *pixels* brancos foram desconsiderados nesta etapa pois, o objetivo era se verificar se a rede era capaz de fornecer as linhas separatórias entre as fácies sísmicas (compostas de *pixels* pretos).

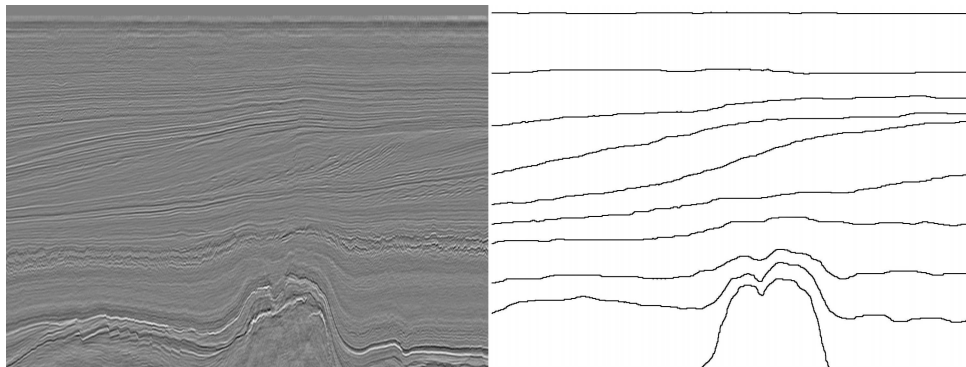
Resumidamente, propomos uma nova arquitetura de rede neural baseada no modelo *encoder-decoder* (DNFS) com a quantidade de parâmetros otimizada para a segmentação de fácies sísmicas e que é treinada em poucos minutos, uma função de perda composta que considera a distribuição espacial dos *pixels* preditos pela segmentação, um modo simples de se adaptar os *datasets* para a segmentação binária de fácies sísmicas, uma função de avaliação utilizada para mensurar as predições da rede proposta e realizamos diferentes experimentos para encontrar as combinações de parâmetros mais adequadas aos nossos objetivos.

Figura 1: Comparação de dados médicos e seções sísmicas. Em ambos os casos, as linhas são usadas para delimitar as regiões de interesse (por exemplo, células e fácies sísmicas).

(a) Dados médicos (esquerda) e máscara correspondente (direita) usada por U-net (RONNEBERGER; FISCHER; BROX, 2015)



(b) Seção sísmica (esquerda) e máscara correspondente (direita) usada neste trabalho



Fonte: Adaptado de Ronneberger, Fischer e Brox (2015).

1.2 Questão de pesquisa

A questão de pesquisa deste trabalho é a seguinte: é possível utilizar redes neurais profundas no modelo *encoder-decoder* para automatizar o processo de identificação de fácies sísmicas inserindo a rede no fluxo de trabalho diário do intérprete? Baseado nessa pergunta a hipótese levantada é que uma nova rede neural enxuta na quantidade de parâmetros e baseada na arquitetura da U-Net seria capaz de realizar essa tarefa, pois uma rede *encoder-decoder* pode ser utilizada para identificar, localizar e segmentar objetos. Assim, com essa hipótese, foram levantados os objetivos gerais e específicos que são apresentados nas Seções 1.2.1 e 1.3.

1.2.1 Objetivo geral

O objetivo geral é conceber uma arquitetura de rede neural capaz de realizar a identificação das divisões entre as interfaces de fácies sísmicas. Para tal foi necessário levar em consideração a quantidade de filtros, camadas internas e a função de avaliação de perda da rede neural que

deveriam ser combinados para se identificar as interfaces entre as fácies sísmicas com pouco tempo de treinamento da rede. Complementarmente à identificação correta das divisões entre fácies sísmicas, objetiva-se a implementação de um recurso computacional que permita que esta atividade seja realizada com o menor tempo possível, atuando no sentido de apoiar o trabalho dos especialistas.

1.3 Objetivos específicos

Os objetivos específicos deste trabalho são listados a seguir:

- Verificar quais tipos de arquiteturas variantes da U-Net e StNet são adequadas a esse tipo de problema;
- Verificar como o conjunto de dados deve ser criado e pré-processado para que a rede neural possa ser treinada em menos de uma hora;
- Identificar quais métricas e funções de avaliação de perda são adequadas a serem aplicadas durante o processo de treinamento;
- Apresentar uma nova estrutura de rede neural junto de métricas e funções de avaliação de perda que levem em consideração características do problema sobre o qual essa dissertação se enfoca.

1.4 Contribuições

Devido à complexidade da realização de identificação de fácies sísmicas, ferramentas que auxiliem esse trabalho e possam ser inseridas no fluxo diário de trabalho de intérpretes sísmicos (geofísicos/geólogos) se fazem necessárias. Para se inserir uma rede neural nesse fluxo, é importante se levar em consideração o tempo de treinamento e a capacidade de generalização. Por isso este trabalho explorou um modo de segmentação binária de múltiplas fácies sísmicas, para que além de auxiliar o intérprete, a rede fosse treinada e fornecesse predições em menos de uma hora.

Assim, as principais contribuições deste trabalho podem ser enumeradas a seguir:

- Introduzimos a DNFS, uma arquitetura baseada na U-Net e StNet com menos parâmetros que para segmentação binária de dados sísmicos. Nosso método é capaz de treinar uma rede neural para segmentar um número arbitrário de fácies sísmicas porque focamos apenas nas transições entre essas fácies.
- Apresentamos uma função de perda composta pela combinação linear de *cross-entropy* e *Jaccard loss*. Também realizamos vários experimentos para otimizar o coeficiente da função de perda, que pondera a combinação de *cross-entropy* e *Jaccard loss*.

- Detalhamos um processo simples de adaptação de conjunto de dados para a segmentação de fácies sísmicas com base na segmentação binária. Usamos linhas pretas para separar fácies sísmicas e há apenas um fundo branco entre essas linhas.
- Realizamos uma extensa avaliação experimental para avaliar o desempenho de nossa abordagem, incluindo ajuste de hiper-parâmetros e uma comparação com U-Net e StNet.
- Apresentamos uma rede neural que pode se treinada em aproximadamente 15 minutos e oferece um índice de acima de 95% relativo à métrica IoU sobre os *datasets* StData-12 (DI; GAO; ALREGIB, 2019a) e Facies-Mark (ALFARRAJ, 2019).

1.5 Publicações

Os resultados desta pesquisa viabilizaram a preparação dos artigos listados a seguir:

1. LIMA, G. et al. Binary segmentation of seismic facies using encoder-decoder neural networks. In: LATINX IN AI RESEARCH (LXAI) WORKSHOP AT NEURIPS, 2020. **Anais...** [S.l.: s.n.], 2020.
2. , G. et al. An encoder-decoder deep neural network for binary segmentation of seismic facies. **Under review.**, [S.l.], 2021.
3. LIMA, G. et al. Deep learning for seismic facies segmentation: a systematic literature review. **In preparation.**, [S.l.], 2021.

No primeiro e segundo artigo da lista acima, foi proposta uma rede neural convolucional profunda para se realizar a segmentação binária de sismofácies. Este trabalho é um *short paper*, o qual foi revisado por pares e selecionado para uma apresentação longa no workshop *Latin X in AI Research*, que aconteceu junto ao NeurIPS 2020. Uma versão estendida deste artigo foi preparada para um periódico e se encontra em revisão. Além desses dois artigos, há uma revisão sistemática da literatura em fase de preparação, cujo foco é a apresentação das técnicas de *Deep learning* utilizadas para a segmentação de fácies sísmicas.

1.6 Estrutura do texto

O texto é estruturado da seguinte forma:

- Capítulo 2: contém a fundamentação teórica, onde são apresentados os conceitos de Geologia voltados à identificação de fácies sísmicas e os de redes neurais, ambos utilizados em outras partes do texto.
- Capítulo 3: apresenta a método utilizado para a busca de trabalhos relacionados, assim como uma breve descrição e discussão dos mesmos.

- Capítulo 4: apresenta o método utilizado no estudo da arquitetura, adaptação dos dados para treinamento da rede neural, a criação da função de perda combinada e o modelo de treinamento utilizado para a rede neural.
- Capítulo 5: apresenta alguns testes preliminares com três arquiteturas de redes neurais e os respectivos resultados. Ambos com o objetivo de se verificar a configuração mínima da rede neural para a segmentação binária das interfaces de separação entre as fácies sísmicas.
- Capítulo 6: contém as considerações finais sobre este trabalho e os trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

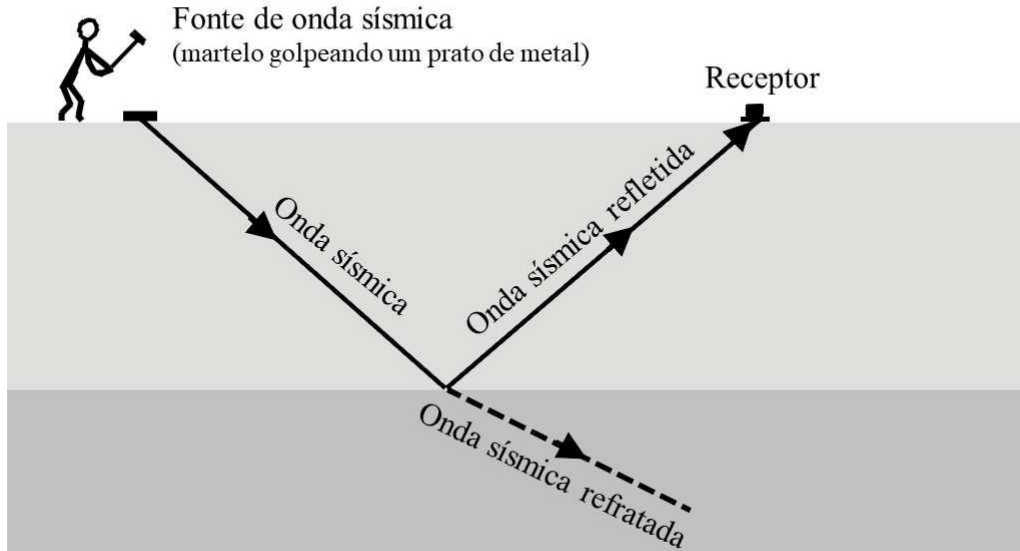
Este capítulo apresenta o embasamento teórico que fundamenta o presente trabalho. Na Seção 2.1 serão abordados conceitos de Geofísica relacionados à identificação de fácies sísmicas e a Seção 2.2 apresenta os conceitos sobre redes neurais que são utilizados neste trabalho.

2.1 Sísmica de reflexão

A sísmica é a base da Sismologia o qual é um ramo da Geofísica que visa a compreensão do interior da Terra através da análise do movimento do solo. Inicialmente, os sismólogos descobriram que as ondas sísmicas geradas pelos terremotos continham informações importantes sobre o interior da Terra (crosta, manto e núcleo). Essas mesmas ondas poderiam ser geradas artificialmente, em um nível menor de intensidade que as provenientes de terremotos, com o intuito de se interpretar a estrutura superficial da terra, localizar minerais, recursos hídricos e petrolíferos. A sismologia se concentra no estudo da geração, propagação e gravação das ondas elásticas na terra e dos recursos que as produzem. Na exploração petrolífera é largamente utilizada para se identificar trapas (*traps*) e otimizar a extração de óleo e gás em reservatórios complexos (IKELLE; AMUNDSSEN, 2018).

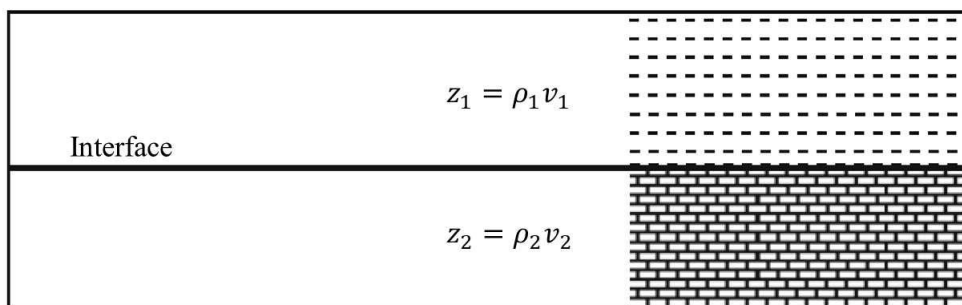
A geração e propagação de ondas sísmicas é feita com a aplicação de uma fonte externa de força sobre uma superfície. Essa força dá origem à deformações que produzem ondas elásticas que se propagam para as subcamadas da região onde a força foi aplicada. Para a geração de ondas sísmicas é necessário a utilização de uma fonte mecânica impulsiva de energia. A fonte pode ser natural (como um terremoto) ou artificial (como canhões de ar ou dinamite) (BAY et al., 2019). Ao ser gerada, a onda causa uma deformação no seu ponto de origem, o qual gera outras deformações adjacentes, as quais carregam consigo uma parte da energia da anterior (DONDURUR, 2018), esse processo se repete até que toda energia inicial da onda tenha se dissipado no meio em forma de calor (NANDA, 2016). Assim, a onda é propagada em todas as direções possíveis. Durante o percurso da onda sísmica para as os meios mais internos de uma área, parte da onda é refletida quando atravessa a interface (Figura 3) entre as camadas rochosas. Essa reflexão é capturada por sensores posicionados na superfície. Cada sensor é responsável por registrar as oscilações da intensidade das reflexões de ondas captadas e o seu tempo de chegada, Figura 4. Cada sensor dá origem a um traço sísmico (conjunto de valores de impedância de onda associados a um sensor) e ao conjunto de traços se dá o nome de seção sísmica (ALSADI, 2016).

Figura 2: Geração de ondas sísmicas.



Fonte: Adaptado de (BAY et al., 2019).

Figura 3: Representação de uma interface entre dois meios diferentes. Impedância acústica: Z_1 e Z_2 . Densidade: ρ_1, ρ_2 . Velocidade de propagação de onda: V_1, V_2 .



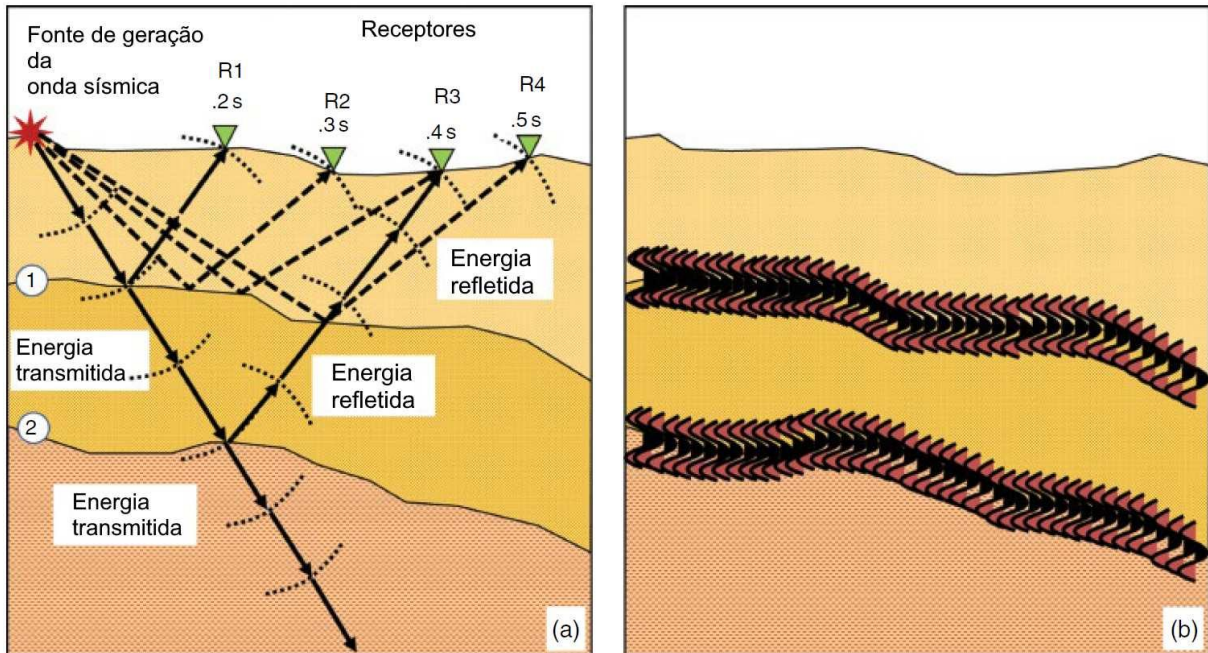
Fonte: Adaptado de (ALSADI, 2016).

A aquisição de dados sísmicos usualmente faz a utilização de centenas de receptores tanto em terra como em mar. Os receptores são organizados equidistantes entre si em uma linha reta e cada receptor dará origem a um traço sísmico. A cada linha se dá o nome de linha sísmica ou seção sísmica. Quando a aquisição de dados sísmico é feita utilizando-se uma única linha se diz que é bidimensional (2D), pois o dado resultante é composto pelo tempo que a onda sísmica levou para percorrer o trajeto até a interface entre as rochas e o valor da magnitude da reflexão da onda. Em contrapartida, nos casos onde muitas linhas sísmicas são utilizadas, a aquisição é dita tridimensional (3D) (THOMAS, 2004).

A Figura 4 exhibe o leiaute de aquisição em terra. Na parte (a), há um diagrama mostrando

o leiaute da energia transmitida e refletida do ponto de disparo e os receptores. Em todas as interfaces, uma parte da energia é transmitida e outra é refletida (estas alcançam os receptores na superfície). Existem duas interfaces refletoras, numeradas (1) e (2).

Figura 4: Leiaute de aquisição em terra. Rn: número do receptor; em (b) traços sísmicos simplificados correspondentes as interfaces em (a). Adaptado de (AYAN; YADAV, 2018)

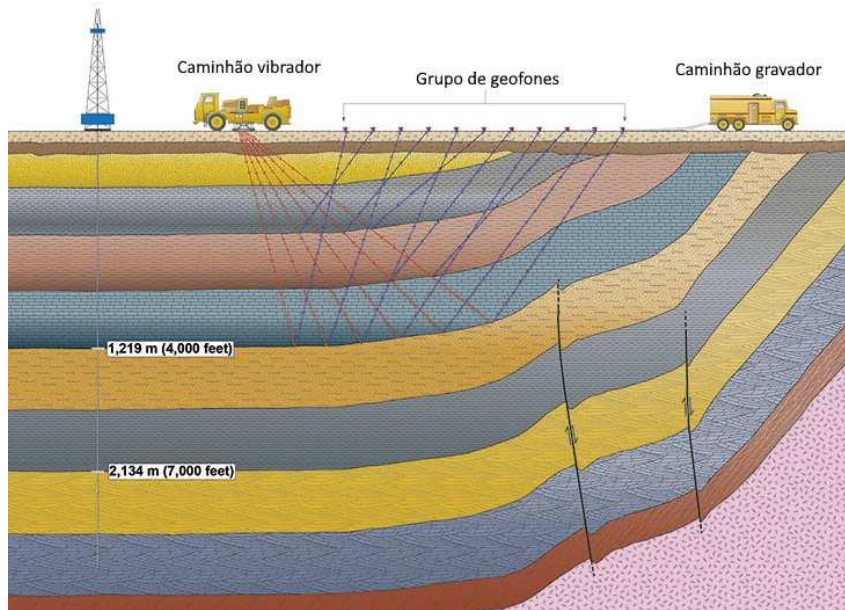


Fonte: Adaptado de (AYAN; YADAV, 2018)

Os ambientes de aquisição podem ser em terra ou em mar. A aquisição de dados sísmicos terrestres (Figura 5) pode ser realizada em qualquer parte do solo terrestre exposto na superfície. Porém, esse tipo de aquisição pode sofrer de problemas de logística como área de difícil acesso ou proibidas, altas atividades que produzem ondas elásticas no solo gerando muito ruído durante a captura dos sensores, etc. As fontes mecânicas de energia comumente utilizadas nessas áreas incluem explosivos (como dinamite ou cordas explosivas), vibradores, canhões de ar e queda de peso. A aquisição marinha, envolve a aquisição de dados sísmicos em locais rasos (10–40 m), ou em águas ultraprofundas (> 1500 m) por navios especializados com matrizes de hidrofones (receptores) rebocadas por um ou mais cabos, chamados *streamers* (Figura 6) (THOMAS, 2004).

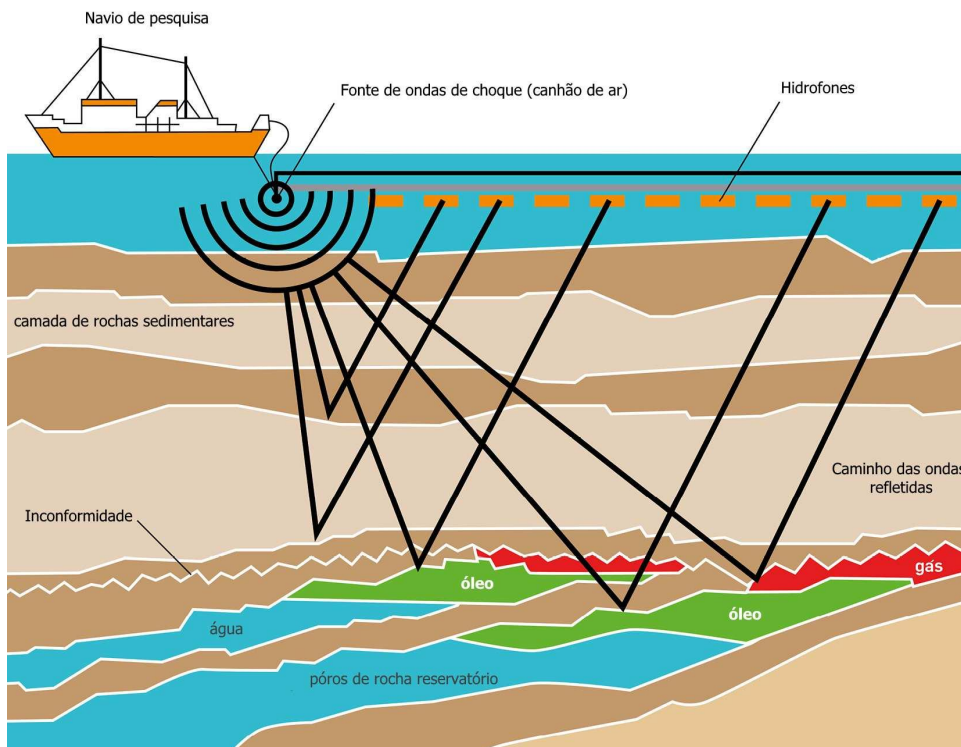
Após a aquisição sísmica, é feito o processamento de dados sísmicos, o qual é uma sequência de processos matemáticos e estatísticos executados nos dados brutos da aquisição, a fim de se extrair informações voltadas a interpretações sísmicas nas quais o geofísico possa identificar estruturas geológicas favoráveis a acumulação de hidrocarbonetos. O objetivo principal é corrigir os sinais sísmicos através da diminuição de ruído (como sinais captados do ambiente), correção do tempo de viagem de reflexão e correção da forma e posição das ondas refletidas (ALSADI, 2016). O resultado do processamento são dados sísmicos chamados de *pos-stack* onde foram feitas melhorias no sinal de ondas sísmicas e correções nas posições dos eventos de reflexão

Figura 5: Esquema de aquisição sísmica em terra.



Fonte: Adaptado de (WYOMING, 2019).

Figura 6: Aquisição sísmica no mar.



Fonte: Adaptado de (KRISENERGY, 2019)

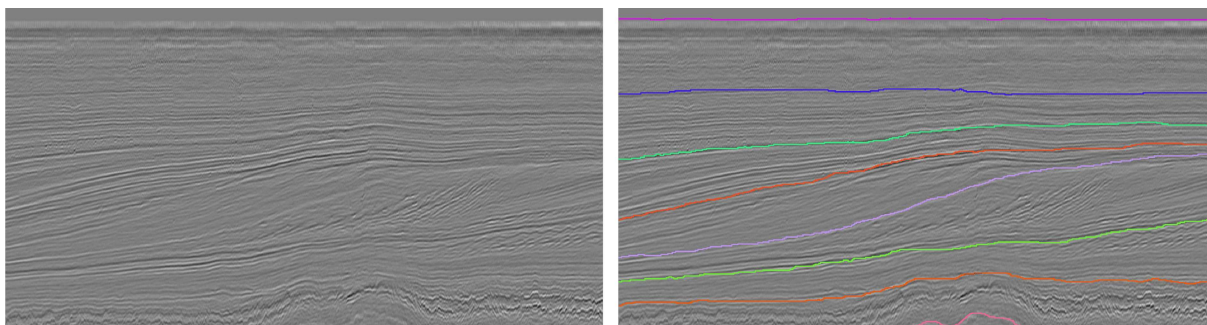
(KEAREY; BROOKS; HILL, 2002).

A fase de interpretação dos dados sísmicos lida com a análise dos dados reflexivos providos

pela sísmica com o intuito de se identificar regiões com algum significado geológico. Para ONAJITE (2013) o objetivo da interpretação é extrair todas as informações de subsuperfície disponíveis dos dados sísmicos processados. Isso inclui informação estrutural, estratigráfica, propriedades de rochas subterrâneas, velocidade de propagação de onda, tensão, etc. Esse processo requer uma aquisição sísmica com alta resolução, conhecimento da geologia local (área de estudo) e experiência do intérprete. Durante a interpretação se verifica a existência de padrões de reflexão para que os mesmos sejam associado a alguma propriedade sísmica. A interpretação é feita através da seleção dos topos (início) dos padrões e da correlação entre esses topos os quais podem formar uma área ou volume.

A interpretação estrutural é comumente realizada durante a fase inicial da exploração de reservatórios de petróleo, onde se verifica se há uma probabilidade alta da existência de hidrocarbonetos que justifique os investimentos monetários a serem realizados. A análise estrutural é o estudo da geometria das interfaces das rochas a partir dos tempos de reflexões (ONAJITE, 2013). A sua principal aplicação é a busca de armadilhas estruturais (traps) que possam conter hidrocarbonetos. Os eventos de reflexão de interesse são marcados utilizando-se cores e rotulados com algum nome, até que seu significado geológico seja estabelecido através da correlação com outros dados, como, poços os quais contém informações de litologia associados a profundidade (MORAIS, 2013). A Figura 7 contém um exemplo de interpretação estrutural.

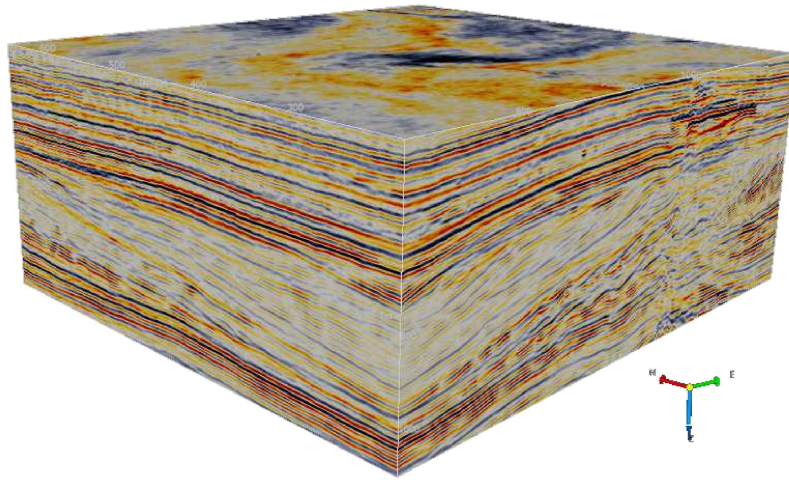
Figura 7: Exemplo de interpretação estrutural. As linhas coloridas representam as geometrias identificadas na seção sísmica.



Fonte: Adaptado a partir do *dataset StData-12* DI; GAO; ALREGIB (2019b)

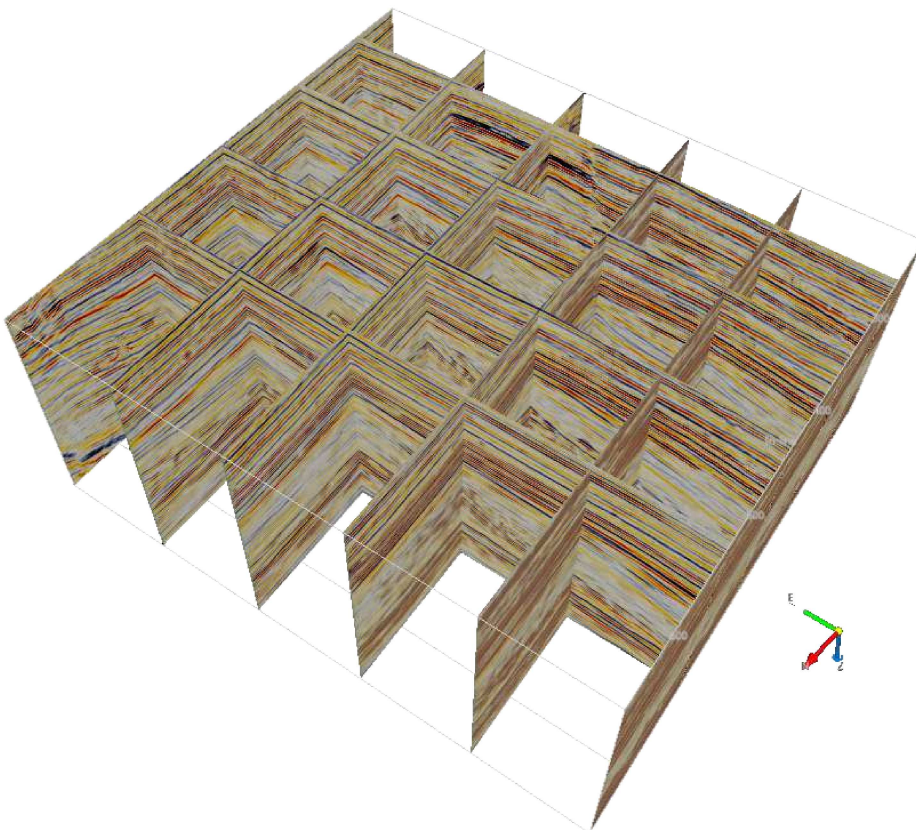
Os dados sísmicos processados podem conter três tipos de seções sísmicas: *inline*, *crossline* e *depth* (ou *time*). A seção *inline* é formada a partir de valores fixos nos eixos Y e Z de um volume sísmico. A *crossline* é formada a partir de valores fixos nos eixos X e Z de um volume sísmico. Já a seção *depth* (ou *time*) é formado a partir de um corte transversal nas *inlines* e *crosslines*. Uma analogia para se entender uma seção sísmica é vê-la como um plano que compõe o volume sísmico, como exibido nas Figuras 8 e 9, onde várias seções se entrecortam.

Figura 8: Exemplo de um volume sísmico.



Fonte: Imagens adaptadas do *dataset F3 Block TERRANUBIS* (2020)

Figura 9: Exemplo de algumas seções sísmicas que compõem o volume da Figura 8.

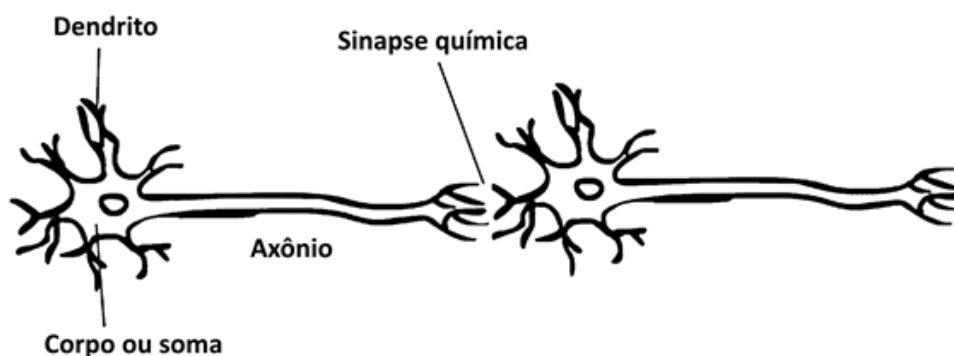


Fonte: Imagens adaptadas do *dataset F3 Block TERRANUBIS* (2020).

2.2 Redes neurais

O cérebro humano é o responsável pelo controle de uma vasta quantidade de atividades complexas realizadas no corpo humano como a capacidade de entender, perceber e interagir com o ambiente através da utilização dos órgãos sensitivos e motores. Os neurônios são os componentes do cérebro responsáveis por essas iterações. Eles se interligam e possuem a capacidade de enviar ou receber informações entre si, e são compostos por três partes: corpo ou soma, dendritos e axônios. O corpo é responsável pelo processamento dos sinais elétricos que são recebidos através dos dendritos. Os dendritos são ramificações de fibras que partem do corpo e recebem os estímulos nervosos advindos de outros neurônios. Quando os estímulos nervosos atingem um determinado grau, o corpo pode gerar novos estímulos nervosos que são enviados a outros neurônios através do axônio. A conexão entre os axônios e dendritos é feita por um processo químico chamado de sinapse química (GRAUPE, 2007). Uma observação a ser feita é que o corpo e os dendritos formam a superfície de entrada de informações e o axônio a superfície de saída, essa característica é importante de ser frisada pois, é utilizada no modelo matemático descrito mais adiante. A Figura 10 apresenta um modelo simplificado de neurônio biológico.

Figura 10: Modelo simplificado de neurônio biológico.

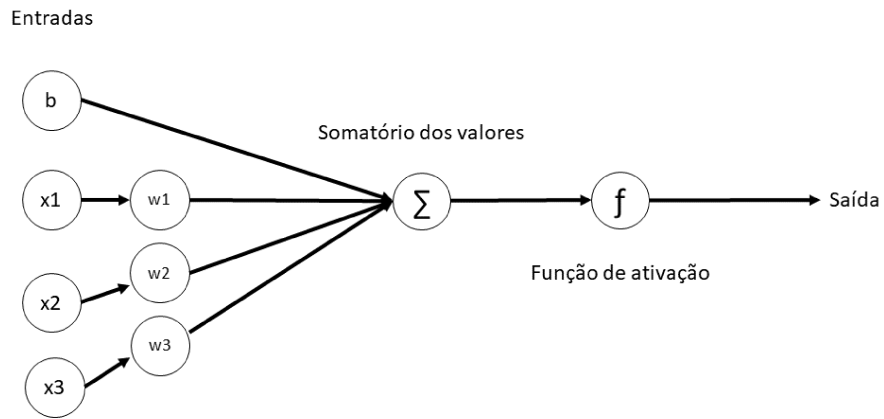


Fonte: Adaptado de Fred (2019).

Baseado na estrutura e funcionamento do neurônio biológico foi criado o neurônio artificial por HAYKIN (2001). O qual é um modelo matemático que busca simular de forma simplificada o funcionamento e estrutura do neurônio biológico. Em termos gerais, podemos definir que o neurônio matemático é um componente que calcula a soma ponderada de valores de entradas, aplica uma função de ativação sobre a soma, e se o valor fornecido pela função for suficiente para que haja a ativação do envio de um sinal, esse resultado é passado adiante. Sua estrutura é composta por uma entrada $(x_1, x_2, x_3, \dots, x_n)$ multiplicada por pesos $(w_1, w_2, w_3, \dots, w_n)$ e somada a um valor constante (b) , o resultado é passado a uma função (δ) que restringe os valores de saída a um intervalo. A Figura 11 mostra o esquema desse modelo de neurônio (também chamado de perceptron) (GRAUPE, 2007).

As redes neurais artificiais (*artificial neural networks* - ANNs) são formadas por conjuntos de neurônios artificiais dispostos em camadas, cada camada é interligada a sua adjacente. Nessas

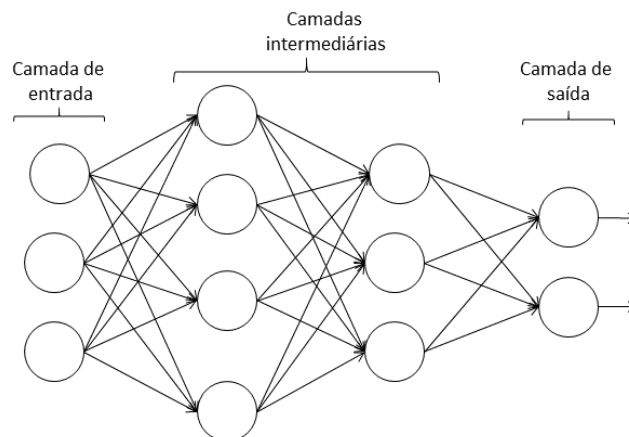
Figura 11: Exemplo de perceptron.



Fonte: Criada pelo autor.

redes a camada c_{i+1} recebe informações da camada c_i (BRAGA; CARVALHO; LUDERMIR, 2000). Há outros modelos de redes onde a informação não flui somente para frente, como as redes recorrentes, mas neste trabalho elas não serão apresentadas. A razão de existir essa organização é que através da mesma é possível a solução de problemas não-lineares. A Figura 12 exibe um modelo de rede neural composto por quatro camadas. A camada mais à esquerda é a de entrada, a mais à direita é a de saída e todas as outras são as intermediárias.

Figura 12: Modelo de rede neural.



Fonte: Criada pelo autor.

A utilização de redes neurais se baseia na ideia da utilização de algoritmos que permitam a um sistema aprender a realizar tarefas a partir de dados e não através de instruções explícitas de como proceder. Esses algoritmos utilizam como entrada um conjunto de informações para criar um modelo que representasse os dados fornecidos. Os modelos teriam a capacidade de realizar alguma tarefa com o conhecimento adquirido a partir do treinamento sobre os dados (MOHRI; ROSTAMIZADEH; TALWALKAR, 2012). Ou seja, as redes extrairiam propriedades/características (*features*) intrínsecas dos dados para poder fornecer predições. Ex.: altura, peso e idade de uma pessoa podem ser consideradas propriedades.

As redes neurais podem ser empregadas na resolução de tarefas como classificação, regressão, ranqueamento, agrupamento, redução de dimensionalidade e segmentação. Na classificação é realizada a associação de classes/rótulos/categorias a objetos (ex.: classificação de tipos de documentos, identificação de quais animais existem em uma imagem, reconhecimento de caracteres, análise de sinais para determinação da fonte e natureza do sinal). Na regressão é feita a predição de valores tendo-se como entrada um ou mais itens (ex.: predição da quantidade de gols em uma partida de futebol). No ranqueamento é realizado o ordenamento de itens de acordo com algum critério (ex.: retorno de páginas web mais relevantes para uma determinada busca). No agrupamento é feito particionamento de itens em conjuntos com características semelhantes (ex.: agrupamento de doenças a partir de sintomas descritos em prontuários médicos). Na redução de dimensionalidade é realizada a transformação da representação de itens em um conjunto menor mas preservando as características principais da representação inicial (ex.: análise da presença de um determinado objeto em uma foto). Na segmentação é feito o reconhecimento e localização de objetos em uma cena ou imagem. Sendo esta a tarefa utilizada neste trabalho, onde realizamos a identificação e localização das interfaces divisórias entre múltiplas fácies sísmicas.

Como mencionado anteriormente, uma rede neural é composta por vários neurônios, pesos e um valor de constante. Durante o treinamento da rede é realizado o ajuste dos pesos e das constantes em várias iterações (chamadas *epochs*) para que ela seja capaz de prever alguma informação. Este processo pode ser realizado de três modos: supervisionado, não-supervisionado e por reforço. No primeiro, a rede neural recebe como entrada valores x e aprende a mapeá-los para valores y , ou seja, se sabe à priori quais informações a rede deve ser capaz de prever (rótulos) devido aos exemplos de entrada e saída esperadas. O treinamento não-supervisionado não requer que haja qualquer conhecimento sobre a saída esperada, a rede extrai informações dos dados de modo a criar grupos nos quais os elementos contidos possuam propriedades semelhantes. Por último, aprendizado por reforço realiza a análise contínua da diferença entre a resposta produzida pela rede e o resultado esperado; a cada ação tomada pela rede, verifica-se quantitativa ou analiticamente qual o resultado no ambiente externo, se for satisfatório, os parâmetros são atualizados para se reforçar o comportamento aprendido (SILVA et al., 2016).

Neste trabalho foi utilizado o processo de treinamento supervisionado, onde as entradas eram imagens sísmicas pós-stack e as saídas imagens contendo máscaras das fácies sísmicas interpretadas. O treino da rede neural foi feito utilizando-se o algoritmo chamado *backpropagation*, o qual

é a aplicação de dois estágios consecutivamente. O primeiro é chamado de *forward propagation*, onde os valores de entrada $(x_1, x_2, x_3, \dots, x_n)$ são propagados camada por camada até que se produza os valores de saídas. Neste estágio o objetivo é obter uma resposta da rede a entrada fornecida a ela, levando-se em conta os valores dos pesos e das constantes atuais da rede, os quais não são modificados neste passo. Antes do início do segundo estágio, o valor de saída da rede é comparado com o valor esperado de saída, a diferença obtida informa o quão distante a rede está de fornecer o valor real esperado para a saída a partir de um conjunto de entrada. Sendo, que a configuração desses estágios leva em consideração que o aprendizado é do tipo supervisionado onde o resultado esperado de saída é conhecido primariamente. No segundo estágio, chamado de *backward propagation* é utilizado o valor do erro calculado para o ajuste dos pesos dos neurônios e das constantes da rede. O treinamento da rede é feito a partir da aplicação desses dois estágios múltiplas vezes (*epochs*), até que se atinja um número pré-determinado de iterações ou o valor do erro calculado esteja em uma faixa aceitável para o problema sendo tratado.

Para calcular o erro fornecido, é utilizada uma função de perda. Tomando uma função de perda $L(\theta)$ sobre o conjunto θ de pesos e constantes, é necessário se calcular como atualizar θ de modo que $L(\theta)$ seja minimizado em relação ao conjunto de entradas x utilizadas para treinamento. Para isso é utilizado o método chamado gradiente descendente, neste método considera-se que os valores assumidos por $L(\theta)$ sejam representáveis em um plano e o objetivo é encontrar o menor valor que poderia ser assumido por $L(\theta)$. O gradiente da função L é definido como uma função derivativa parcial:

$$\nabla L(\theta) = \frac{\partial L(\theta)}{\partial \theta} \quad (2.1)$$

A cada valor calculado do gradiente (o resultado numérico da derivada parcial depende dos valores dos pesos para cada iteração) será definido um valor de atualização dos pesos e das constantes, como se segue abaixo:

$$\theta = \theta - \eta \nabla L(\theta) \quad (2.2)$$

Onde η é chamado de taxa de aprendizado e é utilizado como uma taxa de atualização dos pesos da rede neural durante o aprendizado. Porém, durante o treinamento é comum a utilização de centenas ou milhares de dados de entradas, os quais geralmente não podem ser armazenados na memória RAM do computador. Assim, uma variação do gradiente descendente é utilizado, a qual é chamada gradiente descendente estocástico. Esse método se baseia na utilização de uma quantidade m de amostras aleatórias do conjunto de entrada de tamanho N , onde se calcula o gradiente dessa amostra e ao final é realizada uma média dos gradientes para se atualizar θ (AGGARWAL, 2018).

$$\theta = \theta - \eta/m * \sum^N \nabla L(\theta) \quad (2.3)$$

Após o cálculo do erro é necessário que os pesos da rede sejam atualizados, isso é feito através do algoritmo de *backpropagation*. Esse algoritmo realiza a atualização dos pesos (w) e das contantes (b) através do cálculo do gradiente utilizando derivadas parciais: $\partial L/\partial b$ e $\partial L/\partial w$ (AGGARWAL, 2018). Para cada neurônio que compõe as camadas internas da rede, o seu valor de atualização é a media do quanto esse neurônio contribuiu para o resultado do erro calculado. Esse valor pode ser calculado através da Equação

$$\delta_j^l = \frac{\partial L}{\partial z_j^l} \quad (2.4)$$

onde l representa a n -ésima camada, e z_j^l o j -ésimo neurônio da camada l .

Para os neurônios da camada de saída, o erro pode ser calculado através da Equação

$$\delta_j^l = a_j^l - y_j \quad (2.5)$$

, onde a_j^l é o valor calculado a partir da função de ativação σ para o j -ésimo neurônio de saída e y_j o valor real de saída correspondente.

Após o cálculo do erro ser realizado, este valor é propagado para os neurônios da camada l a partir da camada $l + 1$, como se segue:

$$\delta^l = w^{l+1} \cdot \delta^{l+1} \odot \sigma'(z^l) \quad (2.6)$$

, onde \odot é o produto elemento a elemento e z^l o valor de saída do somatório para a camada l .

Através da combinação das Equações 2.5 e 2.6 e com a utilização da regra da cadeia, chega-se a equação de propagação do erro entre todas as camadas, a qual pode ser definida como:

$$\frac{\partial L}{\partial w_{j,k}^l} = \frac{\partial L}{\partial z_j^l} \frac{\partial z_j^l}{\partial w_{j,k}^l} = a_k^{l-1} \cdot \delta_j^l \quad (2.7)$$

, onde $w_{j,k}^l$ representa o peso da conexão entre k -ésimo neurônio na camada $l - 1$ e j -ésimo neurônio na camada l , b_j^l a constante associada ao j -ésimo neurônio na camada l .

Em resumo, o treinamento da rede neural inclui duas etapas. Na primeira, a informação atravessa a rede da camada de entrada até a camada de saída obtendo-se um valor de predição. Na segunda, é calculado o erro a partir da diferença da predição e do valor esperado como saída através de uma função de perda (*loss function*), essa diferença é propagada no sentido contrário para que cada peso dos neurônios seja atualizado de acordo com a contribuição relativa do seu neurônio ao erro final. Nessas duas etapas é necessário que sejam escolhidos os hiper-parâmetros os quais são os parâmetros definidos antes do processo de treinamento da rede neural e também especificam a estrutura da rede. Entre esses hiper-parâmetros temos o número de camadas, tipos de camadas, função de inicialização dos pesos, funções de ativação (responsável por delimitar a saída numérica dos neurônios presentes em uma camada); função para cálculo de erro; a taxa

de atualização dos pesos (taxa de aprendizagem); número de iterações (*epochs*), quantidade de exemplos a serem agrupados e processados como se fossem um único exemplo pelo algoritmo de aprendizado (*batch size*); os conjuntos de exemplos a serem utilizados para treinamento, validação e testes. Esses são subconjuntos criados a partir da divisão do conjunto de dados em três. O primeiro subconjunto é chamado de treinamento e é utilizado na etapa de treinamento e geralmente contém a maior porcentagem dos exemplos do conjunto de treinamento. O segundo subconjunto é o de validação o qual é composto por exemplos utilizados para se verificar se a rede está conseguindo aprender. Os exemplos contidos nesse conjunto não devem estar presentes no conjunto de treinamento e podem ser utilizados para ajustes dos parâmetros de treinamento da rede. O último conjunto é chamado de teste, e é utilizado para se aferir se a rede é capaz de fornecer predições sobre exemplos que tenham similaridade com os utilizados durante a etapa de treinamento anteriormente mas que nunca foram disponibilizados a ela.

2.2.1 Redes convolucionais

Redes neurais convolucionais (*convolution neural network* – ConvNet/CNN) são variações de redes multicamadas com inspirações biológicas no córtex visual humano. Essas redes são utilizadas em análises de imagens (detecção de rostos, objetos, animais em fotos), análise de séries temporais, carros-autônomos, etc. Comumente, uma rede CNN é composta por várias blocos de extração de propriedades os quais são formados por três componentes principais: camada convolucional, função de ativação e camada de *pooling*. Opcionalmente, pode ser utilizada uma camada de *dropout* após a camada de *pooling*. A camada convolucional extrai as propriedades de uma imagem de entrada aplicando-se um filtro (*kernel*) ao redor da imagem. A convolução é a multiplicação dos valores de pixel da imagem com os valores do filtro para cada local na imagem de entrada. A camada de *pooling* é usada para reduzir a dimensão das propriedades extraídas e a de *dropout* é responsável por eliminar de modo aleatório alguns neurônios e suas conexões para evitar o sobre-ajuste da rede aos dados de treinamento.

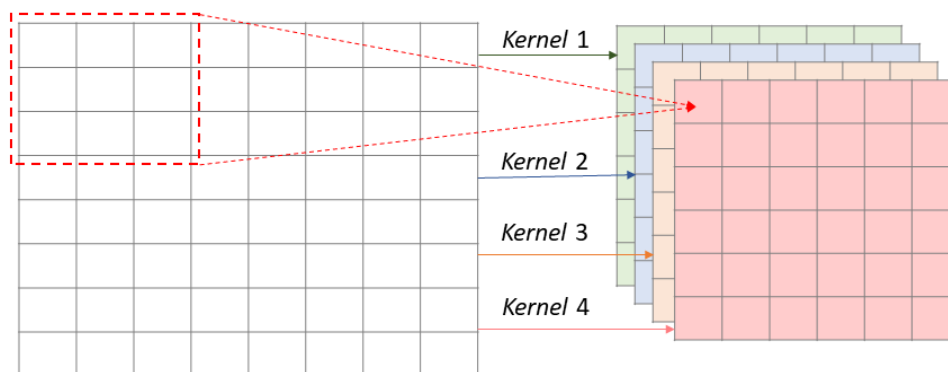
Em uma CNN a camada de convolução tem como finalidade a extração de propriedades da imagem, podendo a rede ser composta por inúmeras camadas convolucionais. As propriedades são identificadas através de filtros chamados *kernels*. A informação de entrada quanto o *kernel* possuem n dimensões e os valores numéricos dos parâmetros armazenados no *kernel*, os pesos das conexões dos neurônios e as constantes são adaptados pelo algoritmo de aprendizagem utilizado para o treinamento. Geralmente, a convolução é aplicada em mais de um eixo ao mesmo tempo da imagem. A extração de diferentes propriedades utiliza *kernels* de tamanho $P \times Q$ sobre a entrada de tamanho $W \times H$ na camada convolucional. A aplicação dos *kernels* resulta em mapas de propriedades que são empilhados e podem servir como entrada para uma próxima camada convolucional ou como saída da rede neural. O mapa de propriedades pode ser definido

pela seguinte Equação:

$$x_{ij}^{lm} = \sigma \left(b^{lm} + \sum_m \sum_{p=0}^{P-1} \sum_{q=0}^{Q-1} w_p q^l m x_{(i+p)(j+q)}^{(l-1)m} \right) \quad (2.8)$$

, onde x_{ij}^{lm} representa o neurônio na camada l_m na posição (i, j) no mapa de propriedades m_{th} criado a partir da aplicação de um *kernel*, m o número de neurônios da camada $l - 1$ e n da camada l . A Figura 13 apresenta um exemplo de mapas de propriedades criados a partir da aplicação de quatro filtros.

Figura 13: Mapas de propriedades criados a partir da aplicação de quatro filtros.



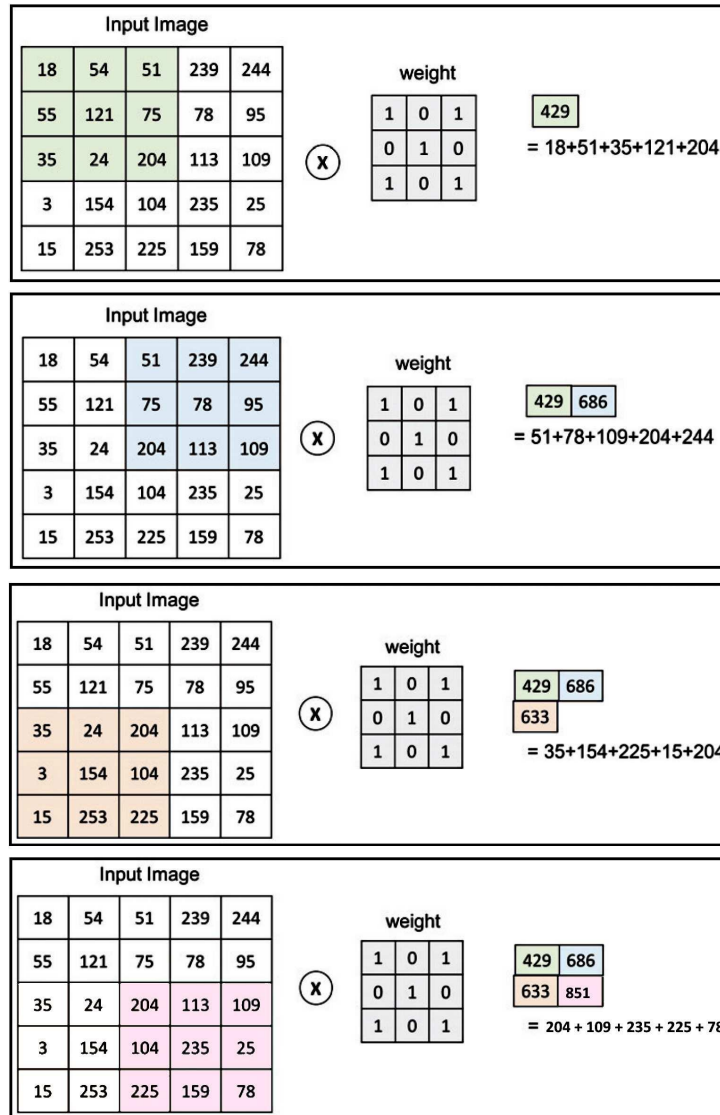
Fonte: Criada pelo autor.

Há dois parâmetros importantes para a aplicação do *kernel* sobre a imagem: *stride* e *padding*. O primeiro se refere ao deslocamento vertical ou horizontal do filtro sobre as células do *grid* que formam a imagem, o segundo se refere a quantidade de células com valores zero a serem adicionadas à borda da imagem original (às vezes é necessário realizar essa operação pois há perda da informação de borda, já que os valores laterais estarão poucas vezes presentes no processo de convolução, para a segmentação de objetos é comum o uso do valor zero para *padding*, o que resulta em uma imagem de saída do mesmo tamanho da de entrada). A Figura 14 exibe a aplicação da operação de convolução sobre o *grid* de uma imagem 2D. Para imagens ou dados de entrada de D dimensões, o *grid* de entrada é dado pelo tamanho $W \times H \times D$ e o *kernel* por $Q \times P \times D$. O parâmetro D é chamado de profundidade (*depth*).

Para introduzir a não linearidade na rede, ou seja, para que a rede contemple problemas mais complexos, é necessário o uso de uma função de ativação não-linear. Nas CNNs, após cada camada convolucional, o mapa de propriedades resultante é passado para uma camada não-linear. A função de ativação ReLU é comumente utilizada para esse fim e segundo (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) o tempo de treinamento da rede pode ser altamente reduzido.

Após a camada de ativação, pode ser utilizada a camada de *pooling* é aplicada após a camada convolucional, o objetivo é reduzir progressivamente o tamanho espacial dos mapas de propriedades com o intuito de diminuir o número de parâmetros e o volume de processamento computacional para a rede. Para imagens coloridas, o *pooling* é feito independentemente em

Figura 14: Aplicação de convolução sobre uma imagem 2D.



Fonte: Criada pelo autor.

cada canal de cor. O *pooling* permite que as propriedades extraídas se tornem aproximadamente invariantes a pequenas mudanças na imagem de entrada. Invariância a deslocamento significa que se a imagem de entrada for deslocada por uma pequena quantidade, os valores da saída providos pelo *pooling* não sofreriam mudanças. Invariância ao deslocamento é uma propriedade importante em situações onde é importante saber se uma determinada propriedade está presente na imagem sem necessariamente ter sua localização (GOODFELLOW; BENGIO; COURVILLE, 2016). Por exemplo, quando se deseja saber se uma imagem contém um rosto ou não, não é necessário saber a exata localização dos *pixels* dos olhos, somente é necessário saber que há um olho no canto esquerdo e um olho no canto direito. Porém, em outros contextos pode ser importante preservar a posição de alguma propriedade a ser mapeada na entrada. Por isso, a escolha do tipo de agrupamento é importante. O algoritmo de *pooling* chamado *max-pooling* é utilizado para se fornecer de uma representação das características mais proeminentes na

imagem (GOODFELLOW; BENGIO; COURVILLE, 2016). A Figura 15 exibe um exemplo da aplicação do *max-pooling*.

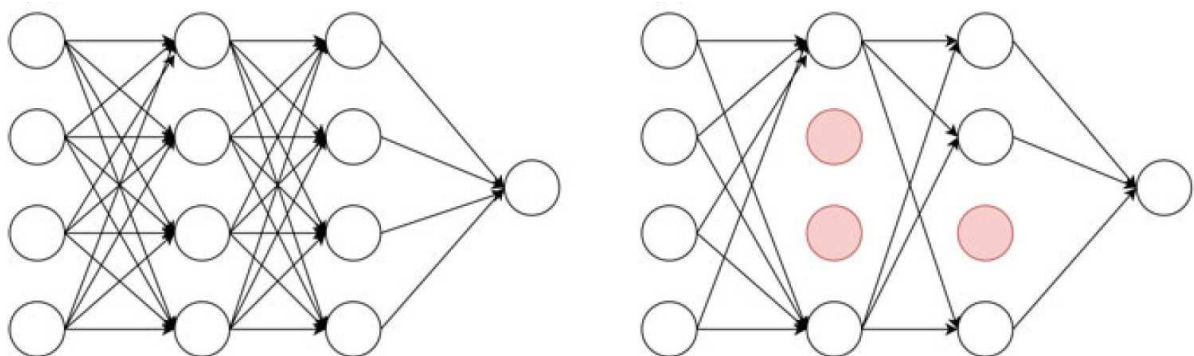
Figura 15: Aplicação de *max-pooling* sobre um mapa de propriedades.

309	291	430	832		
693	498	918	872	693	918
633	751	851	505	933	851
713	933	576	640		

Fonte: Criada pelo autor.

Como as redes neurais profundas consistem de várias camadas ocultas, o que permite que problemas não-lineares sejam aprendidos e durante o treinamento pode haver o sobre-ajuste (*overfitting*) da rede, ou seja, a rede é treinada de modo que é capaz de realizar previsões baseadas nos dados de treinamento mas sobre novos dados essa capacidade é perdida; mesmo esses dados sendo semelhantes ao utilizados para o treinamento. Para mitigar o problema do sobre-ajuste, a técnica de *dropout* (SRIVASTAVA et al., 2014) pode ser introduzida no modelo em que alguns neurônios e suas conexões são temporariamente eliminados durante o treinamento. A rede durante uma iteração e é treinada sem esses neurônios e na próxima iteração $e + 1$ os mesmos são reativados com seus respectivos pesos. Assim, o *dropout* tem a capacidade de reduzir o ajuste excessivo, o que acarreta no aumento da capacidade de generalização da rede sobre novos dados. A Figura 17 apresenta um exemplo de aplicação da técnica *dropout*.

Figura 16: Exemplo da aplicação da técnica de *dropout*.

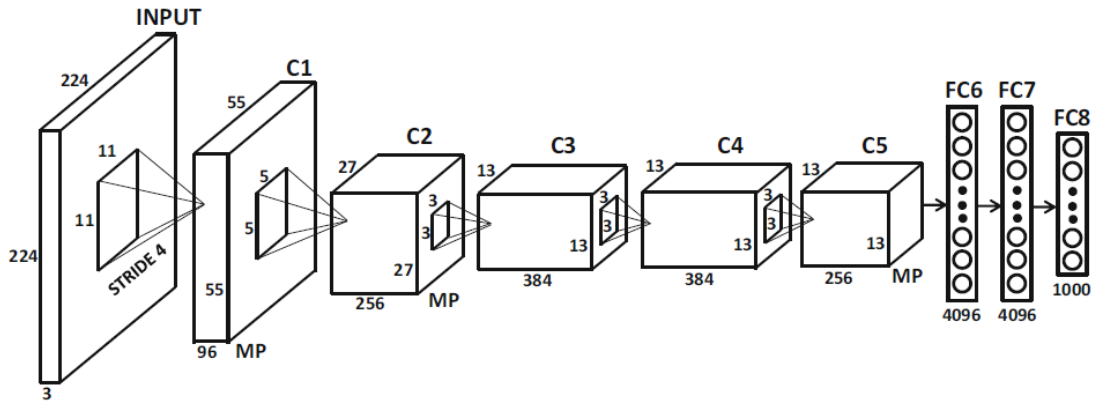


Fonte: Adaptado de Goodfellow, Bengio e Courville (2016).

Para um problema de classificação, após a extração de propriedades através das camadas convolucionais e de *pooling*, o último estágio é a utilização de uma ou mais camadas totalmente conectadas, seguidas pela aplicação da função de ativação *softmax* (HEATON, 2015). Estas possuem seus neurônios conectados com todos os neurônios da camada anterior. Através da função *softmax* é calculada a probabilidade das propriedades extraídas pela CNN pertencerem a

uma determinada classe. A soma de todos os valores preditos para as classes é igual a um e a classe que tiver associada a si o maior valor de probabilidade é utilizada como resultado final. A Figura 17 apresenta um exemplo de rede neural com uma camada totalmente conectada ao final.

Figura 17: Exemplo de rede com camada totalmente conectada.

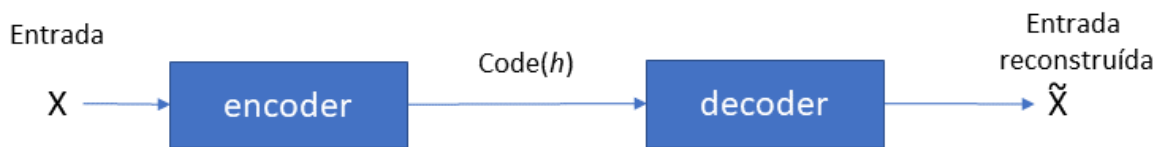


Fonte: Adaptado de Aggarwal (2018).

2.2.2 Arquitetura *encoder-decoder*

Encoder-decoder (EDs) são capazes de aprender representações dos dados de entrada sem nenhuma supervisão, ou seja, o conjunto de treinamento não é rotulado. O objetivo do treinamento de um *encoder-decoder* é extrair um conjunto de propriedades representativas dos dados de entrada e ser capaz de realizar a sua reconstrução em algum formato. A diferença entre uma rede MLP e um ED é que a primeira é treinada para prever um valor Y dada uma entrada X , enquanto, a segunda é empregada para reconstruir a entrada X em algum formato \tilde{X} . O ED é composto por duas partes: *encoder* e *decoder*. O *encoder* é responsável por extrair as propriedades mais importantes na entrada X , o *decoder* é utilizado para reconstruir a entrada X a partir das propriedades extraídas. Durante o treinamento de um ED, as otimizações dos parâmetros pelo algoritmo de treinamento são feitas de modo a minimizar a diferença entre a entrada X e a reconstruída \tilde{X} (WANI et al., 2019). Para a segmentação de imagens, ambas as partes do DE devem possuir a mesma dimensionalidade de saída pois o objetivo é gerar uma imagem que tenha as mesmas dimensões da de entrada. A Figura 18 exibe o processo geral de um AE.

Figura 18: Processo geral de um *encoder-decoder*.



Fonte: Adaptado de Wani et al. (2019).

A imagem 19 exibe a estrutura básica de um ED. Os valores $\{x_1, x_2, \dots, x_n\}$ representam a entrada X e $\{\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n\}$ os valores aproximados para a saída \tilde{X} .

A parte codificadora (*encoder*) pode ser representada por:

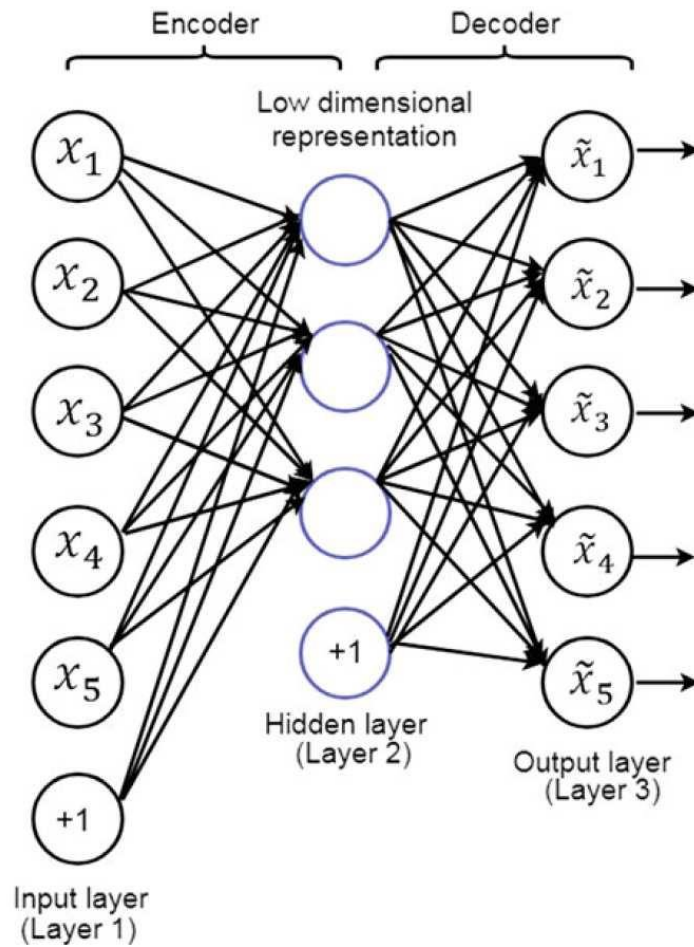
$$a = f(W^{(1)}x + b^{(1)}) \quad (2.9)$$

onde $W^{(1)}$ e $b^{(1)}$ são os pesos e constante da camada do *encoder*, x os dados de entrada e f uma função de ativação para esta camada.

$$h_{(W,b)}(x) = g(W^{(2)}a + b^{(2)}) \quad (2.10)$$

onde $h_{(W,b)}(x)$ representa um vetor com as propriedades extraídas para reconstrução, $W^{(2)}$, $b^{(2)}$, g são os pesos, *bias* e função de ativação do *decoder*.

$$L(x, g(f(x))) \quad (2.11)$$

Figura 19: Processo geral de um *encoder-decoder*.

Fonte: Adaptado de Wani et al. (2019).

onde L é uma função de perda a qual se busca minimizar durante o processo de treinamento através da penalização de $g(f(x))$ por fornecer resultados distantes a X .

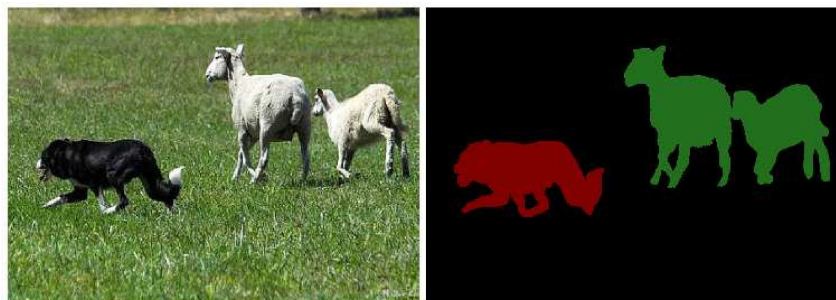
2.3 Segmentação com redes neurais

Nesta seção são apresentados os conceitos utilizados neste trabalho sobre redes neurais. Inicialmente, é feita uma visão geral destes, depois, como a rede neural proposta nesse trabalho foi utilizada, onde mencionamos o tipo de treinamento realizado baseado no Gradiente descendente e no algoritmo de *Back-propagation* assim como as funções de ativação, métricas de comparação e cálculo de erros utilizadas. Na segunda parte sobre redes convolucionais apresentamos os componentes bases que foram um rede convolucional e a arquitetura de rede *encoder-decoder* que é utilizada neste trabalho.

2.3.1 Abordagens de segmentação semântica com redes neurais

A segmentação de imagens é um processo crítico na visão computacional o qual envolve a divisão de uma entrada em segmentos para simplificar a sua análise. Segundo GONZALEZ; WOODS (2018) o problema fundamental na segmentação é particionar uma imagem em regiões que satisfaçam à algum critério. Essas regiões representam objetos ou partes dos mesmos e compreendem conjuntos de *pixels*, ou *super-pixels*, os quais são rotulados de forma que cada pixel compartilhe certas características como cor, intensidade, textura, etc de acordo com algum critério pré-estabelecido. Segundo SULTANA; SUFIAN; DUTTA (2020) a segmentação semântica é rotulação de cada pixel da imagem com um rótulo específico para cada tipo de objeto, se na mesma imagem existir mais de objeto do mesmo tipo, todos serão identificados a partir do mesmo rótulo. A Figura 20 apresenta um exemplo de segmentação semântica. Esse tipo de segmentação é adequado a este trabalho pois permite localizar as áreas onde há fácies sísmicas e identificar cada pixel que as compõem. Um outro motivo é que uma mesma fácies sísmica pode estar localizada em diferentes regiões do dado sísmico, então, o rótulo associado a ela deveria ser único, o que é possível com uso deste tipo de segmentação.

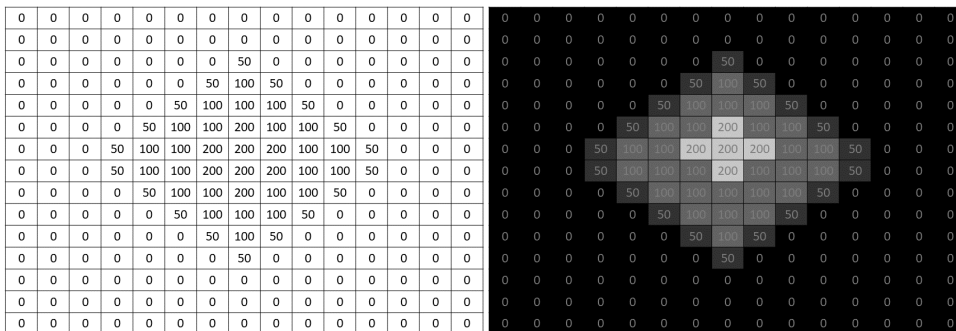
Figura 20: Segmentação de três tipos de objetos: grama, cachorro e ovelha. Em verde está segmentada a ovelha, em vermelho o cachorro, em preto a grama.



Fonte: Adaptado de Liang et al. (2018).

Antes de abordamos as formas de segmentação semântica através de redes neurais, é necessário se definir a estrutura base de imagem utilizada nesse trabalho. Segundo SOLOMON;

Figura 21: *Grid* representativo de uma imagem em escala de cinza. À esquerda os valores da escala de cinza compoendo o *grid*, à direita a imagem criada a partir do *grid*.



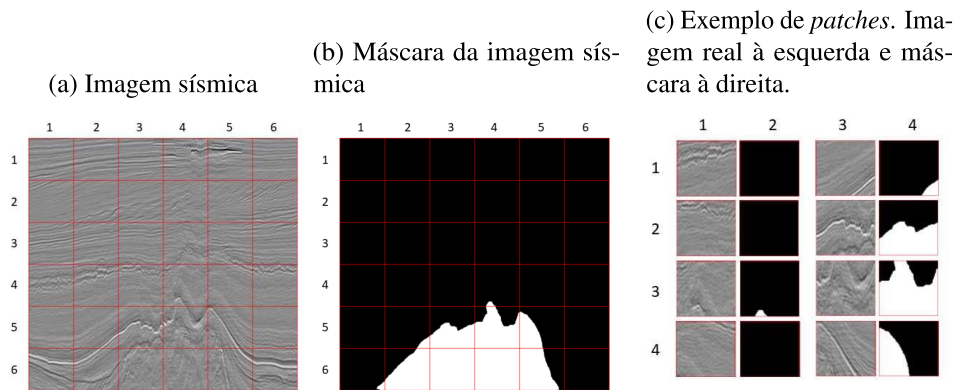
Fonte: Criado pelo autor.

BRECKON (2011) uma imagem pode ser representada por um *grid* com n linhas e m colunas. Cada posição do *grid* seria chamado de *pixel* o qual possui uma localização i para a linha e j para a coluna do *grid* e um valor associado. A Figura 21 apresenta uma imagem 2D em escala de cinza. À esquerda da imagem há um *grid* onde suas posições contém a intensidade de pixel, que na escala de cinza podem estar no intervalo de 0 a 255. À direita, é representada a imagem em escala de cinza referente ao *grid*. O formato apresentado é o utilizado nesse trabalho como entrada e saída para o treinamento das redes neurais, já que a entrada é uma imagem sísmica em escala de cinza e a saída uma imagem contendo uma linha divisória entre as fácies sísmicas, o que será visto com mais detalhes na Seção 4.3.

A segmentação semântica através de redes neurais pode ser feita através da classificação de *patches* extraídos de uma imagem, onde ao *patch* é associado a um único rótulo, o que podemos chamar de *patch-classification* ou PC. Ou através da classificação individual dos *pixels*, onde aqueles que fazem parte de um mesmo objeto recebem o mesmo rótulo, o qual nos referimos nesse trabalho como *pixel-to-pixel classification* ou PPC. Alguns trabalhos analisados como os de (GUO et al., 2020; ZENG; JIANG; CHEN, 2019; DRAMSCH; LüTHJE, 2018) fizeram uso da abordagem de *patch-classification*, já os (LI et al., 2021; XU; LU; WANG, 2019; ZHAO; MUKHOPADHYAY, 2018) realizam o uso da abordagem de *pixel-to-pixel classification*.

A Figura 22 apresenta um exemplo de divisão de uma imagem em *patches* que podem ser utilizados para *patch-classification* ou *pixel-to-pixel classification*. Nesta figura a imagem mais à esquerda representa uma imagem sísmica que contém um domo de sal. A imagem mais à direita apresenta a máscara de segmentação da imagem sísmica. A cor preta representa a área onde não há um domo de sal, já a cor branca representa a área em que há a presença de um domo de sal. Os quadrados delimitados pelas linhas vermelhas representam os *patches* que poderiam ser criados a partir dessas imagens. Na vertical temos a representação das linhas e na horizontal as colunas, ambas são numeradas de 1 a 6. Podemos tomar como exemplo os *patches* da linha 1, todos seriam rotulados como não pertencentes a um domo de sal. Já os *patches* das posições (5, 3), (5, 4), (5, 5), (6, 2), (6, 3), (6, 4), (6, 5) poderiam ser rotulados como pertencendo a uma domo de sal, sendo o primeiro termo representante da linha e o segundo da coluna.

Figura 22: Exemplo de formato para criação de *patches* que podem ser utilizado para *patch-classification* ou *pixel-to-pixel classification*. As linhas vermelhas delimitam a áreas dos *patches*. Na horizontal estão representadas as colunas e na vertical as linhas. A cor branca representa a área onde há um domo de sal, já a cor preta a área onde não há. a) Imagem sísmica e a representações de *patches* quadrados; b) máscara da imagem sísmica contendo um domo de sal; c) Colunas 1 e 3 representam os *patches* das imagens sísmica e as colunas 2 e 4, as máscaras correspondentes.



Fonte: Criado pelo autor.

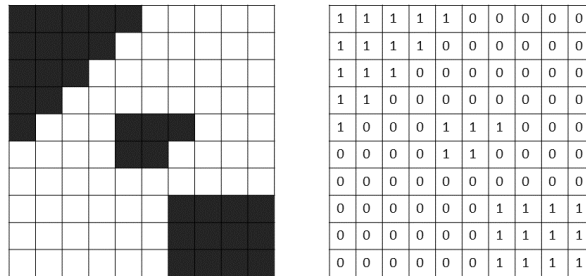
O uso da segmentação baseada em *patch-classification* possui o problema de que identificação de a qual classe o *patch* pertenceria. No exemplo da Figura 22b o *patch* (5, 5) poderia ser rotulado como pertencente a um domo de sal ou não dependendo do critério utilizado. Caso o critério seja que 70% dos *pixels* sejam da cor branca para se rotular como um domo de sal, o *patch* (5, 5) não obteria este rótulo, apesar de conter uma área significativa com a cor branca. Este problema de rotulagem pode ser agravado ao se reduzir as dimensões dos *patches*, o que pode gerar um aumento do número de classificações errôneas.

Na abordagem *pixel-to-pixel classification* os *patches* são rotulados *pixel a pixel*, ou seja, a cada *pixel* da imagem é atribuído um rótulo. A Figura 23 exibe um exemplo de rotulagem. Nas posições dos *pixels* onde havia a cor preta foi inserido o rótulo 1 (um), já na posições com cor branca foram inseridos o rótulo 0 (zero). Assim, a rede neural deverá prever o rótulo a ser associado a cada posição. Esta abordagem foi a escolhida para a execução deste trabalho pois é a que oferecere as predições com maiores acurácias quando comparadas a abordagem de *patch-classification*.

2.4 Considerações

Neste capítulo apresentamos na Seção 2.1 os aspectos de sísmica de reflexão que são utilizados nesta dissertação onde nos focamos em conceitos bases sobre fácies sísmicas e o modo como as suas interpretações são realizadas por profissionais especializados. Na Seção 2.2 estão os conceitos sobre redes neurais, onde apresentamos a arquitetura *encoder-decoder*. Através

Figura 23: Exemplo de rotulação de *pixels* de um *patch*. À esquerda se encontra o *patch*. À direita a rotulação do *patch* onde em cada posição onde há a cor preta foi inserido o rótulo 1 (um) e nas de cor branca o rótulo 0 (zero).



Fonte: Criada pelo autor

dessa arquitetura, identificamos as regiões de transição entre fácies sísmicas. Para isso, nos baseamos na segmentação semântica *pixel-to-pixel*, conforme é apresentada na Seção 2.3, onde a cada *pixel* é atribuída uma determinada classe.

A arquitetura *encoder-decoder* foi escolhida depois de analisarmos trabalhos relacionados (Capítulo 3) que realizam a segmentação de fácies sísmicas através de técnicas de *Deep learning* e verificarmos que a imagem de saída construída por essa arquitetura poderia ser guiada para conter somente as transições das fácies. Essas transições, foram representados de modo binário nessa dissertação, onde a rede neural proposta chamada DNFS (Seção 4.1), teve a função de classificar um *pixel* como sendo pertencente a classe 0 (zero) para áreas entre os limites das fácies sísmicas e 1 (um) como representante das linhas separatórias dessas fácies (Seção 4.3).

A segmentação binária nos permitiu identificar múltiplas fácies sísmicas simultaneamente e diminuir a quantidade de parâmetros necessários para realizar essa tarefa e, conseqüentemente, reduziu o tempo de treinamento da rede, conforme é demonstrado na Seção 5.1. Esse modo de segmentação nos permitiu um diferencial em relação a maioria dos trabalhos relacionados que se focam na segmentação de somente um tipo de fácies sísmicas por vez, conforme é apresentado no Capítulo 3.

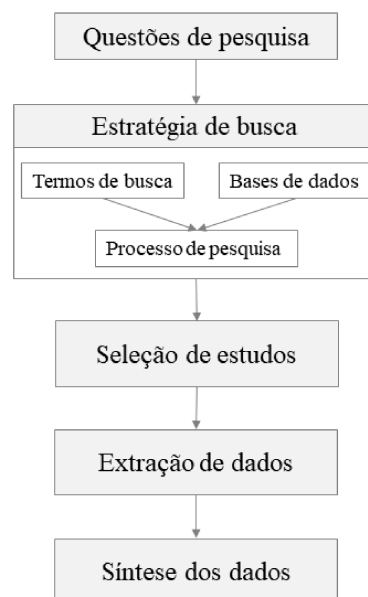
3 TRABALHOS RELACIONADOS

Este capítulo apresenta uma revisão da literatura sobre interpretação sísmica utilizando redes neurais. Optamos por realizar tal revisão para ter uma visão abrangente (abrangente) sobre como redes neurais foram aplicadas sobre dados sísmicos para a identificação de fácies sísmicas. Assim, propor uma rede neural para auxiliar no fluxo de trabalho diário do intérprete na interpretação de fácies sísmica e também em como um *dataset* poderia ser representado para que múltiplas fácies fossem identificadas simultaneamente.

3.1 Protocolo da revisão de literatura

Realizamos uma revisão da literatura aplicando o processo de Revisão Sistemática da Literatura (RSL) proposto por KITCHENHAM; CHARTERS (2007) e seguimos as etapas do protocolo de revisão utilizado por WEN et al. (2012), o qual teve como objetivo analisar sistematicamente modelos de *Deep learning* (estas etapas são exibidas na Figura 24). Na etapa 1, Questões de pesquisa (Seção 3.2) são definidas as questões de pesquisa a serem verificadas com a revisão; Na etapa 2, Estratégia de busca, (Seção 3.3), é apresentada a forma como os termos de busca foram definidos e quais bases de dados foram selecionadas; Na etapa 3, Seleção de estudos (Seção 3.4) apresenta o processo de pesquisa de trabalhos relacionados, onde estes passaram pela verificação dos critérios de exclusão/inclusão e pela análise de qualidade onde recebem uma pontuação; Na etapa 4, Extração de dados, (Seção 3.5), é feita a extração de informações dos trabalhos através de um formulário guia; Na etapa 5, Síntese dos dados (Seção 3.6) as informações coletadas são analisadas.

Figura 24: Estágios do protocolo de revisão.



3.2 Questões de pesquisa

Nesta seção apresentamos o problema, assim como as questões de pesquisa que pretendemos verificar com essa revisão.

O *problema* a ser pesquisado é o seguinte:

- Como se identificar sismofácies em dados de sísmica de reflexão sem a aplicação de atributos sísmicos para explicitar características de fácies sísmicas?

A partir do problema apresentando, investigamos um escopo limitado do mesmo, onde nos focamos em como as técnicas de inteligência artificial baseadas em *Deep learning* foram utilizadas. Assim, o principal objetivo desta revisão é resumir e esclarecer tendências, métricas, benefícios e possíveis técnicas e arquiteturas de *Deep Learning* que foram abordadas na identificação de sismofácies. Com isso em mente, formulamos as questões de pesquisa da Tabela 1 com base no problema citado no parágrafo anterior.

Tabela 1: Questões de pesquisa

Questão de pesquisa	Descrição e motivação
QP1: As abordagens de <i>Deep Learning</i> para identificação de fácies sísmicas são vantajosas em comparação com as abordagens que não a utilizam ?	Visamos identificar se a aplicação de ferramentas baseadas em <i>Deep Learning</i> contribuem para melhorar os softwares de interpretação de fácies sísmicas. Isso é relevante, pois as metodologias atuais necessitam de profissionais especializados e demandam uma maior quantidade de tempo para serem aplicadas devido ao aumento do número de dados sísmicos (TSCHANNEN et al., 2020).
QP2: Quais técnicas de <i>Deep Learning</i> são usadas atualmente para identificar sismofácies?	Estamos interessados em saber quais algoritmos ou arquiteturas de redes neurais foram testados em dados sísmicos para a identificação de sismofácies.
QP3: Quais são as métricas mais usadas em algoritmos e arquiteturas para se identificar fácies sísmicas?	Visamos catalogar quais otimizadores e valores de seus parâmetros foram utilizados para o treinamento das redes neurais. Assim, nos interessou saber quais artigos atuais abordam essa etapa. Além disso, para comparar algoritmos e arquiteturas, é importante definir métricas que medem o desempenho.
QP4: Quais são os esquemas de representação dos dados sísmicos mais utilizados nas predições realizadas através de <i>Deep Learning</i> para se identificar fácies sísmicas?	Com essa questão visamos identificar como as informações de fácies foram utilizadas como entrada para o treinamento de redes neurais.

Fonte: Criada pelo autor.

3.3 Estratégias de busca

Esta seção apresentada a forma como os termos de busca foram definidos e quais bases de dados foram escolhidas.

Foram extraídas a partir do problema e das questões de pesquisa apresentadas nas Seções 1.1 e 3.2 palavras-chave na língua inglesa a serem utilizadas em buscas nas fontes primárias nas bases de dados (Seção 3.4).

As palavras-chave extraídas são exibidas na Figura 25:

Figura 25: Palavras-chave para a realização de buscas em bases de dados.

seismic facies

and

interpretation OR identification OR tracking

and

deep learning OR neural network

Fonte: Criada pelo autor.

As fontes de busca primária utilizadas foram as seguintes bases:

- *IEEE Xplore* - www.ieeexplore.com
- *ScienceDirect* - www.sciencedirect.com
- *Springer* - <https://link.springer.com>
- *ACM* - <https://dl.acm.org/>
- *SEG Library* - <https://library.seg.org/>
- *Google scholar* - www.scholar.google.com

As chaves de buscas utilizadas para cada base estão presentes na Tabela 2:

Tabela 2: Consultas feitas nas bases de dados e períodos de filtragem.

Base	Consulta	Período
<i>IEEE Xplore</i>	<p>(("All Metadata":seismic OR "All Metadata":seismic facies OR "All Metadata":seismic images) AND ("All Metadata":interpretation OR "All Metadata":identification OR "Al Metadata":tracking) AND ("All Metadata": deep Learning OR "All Metadata":neural network)</p>	2010 - 2020
<i>ScienceDirect</i>	<p>(seismic facies) AND (interpretaion OR identification OR tracking) AND (Deep Learning OR neural network)</p>	2010 - 2020
<i>Springer</i>	<p>(seismic facies) AND (interpretaion OR identification OR tracking) AND (Deep Learning OR neural network)</p>	2010 - 2020
<i>ACM</i>	<p>[[All: seismic facies] AND [[All:interpretation] OR [All: tracking] OR [All: identification]] AND [[All: neural network] OR [All: deep learning]] AND [Publication Date: (01/01/2010 TO 12/31/2020)]</p>	2010 - 2020
<i>SEG Library</i>	<p>(seismic facies) AND (interpretation OR identification OR tracking) AND (Deep Learning OR neural network)</p>	2010 - 2020
<i>Google Scholar</i>	<p>(seismic facies) AND (interpretation OR identification OR tracking) AND (Deep Learning OR neural network)</p>	2010 - 2020

Fonte: Criada pelo autor.

3.4 Seleção de trabalhos

Esta seção apresenta os critérios de exclusão e inclusão de trabalhos e os de qualidade, que são utilizados para se pontuar os trabalhos selecionados. Ao final da desta seção, é exibido um sumário com o número de trabalhos em cada etapa da seleção.

3.4.1 Critérios de inclusão e exclusão

O objetivo da definição de critérios é o de identificar trabalhos primários que forneçam evidências diretas para a resposta das questões de pesquisa e reduzir a probabilidade de viés (KITCHENHAM; CHARTERS, 2007). Os trabalhos primários são os que propõem algum estudo e/ou apresentem alguma validação empírica de suas contribuições. Enquanto, estudos secundários seriam os que realizam a revisão sobre um tópico (como revisões sistemáticas).

Os estudos elegíveis a inclusão eram aqueles publicados entre os anos de 2010 e 2020 e que apresentavam alguma contribuição ao campo de interpretação de fácies sísmicas através de técnicas de *Deep learning*. A decisão por esse período se deveu ao fato de que a partir do ano de 2018 essa área passou a receber atenção maior dos pesquisadores, o que pode ser confirmado a partir da análise do gráfico apresentado na Figura 27.

Os trabalhos eram excluídos se fossem secundários, duplicados, não escritos na língua inglesa, trabalhos de literatura cinzenta (por exemplo, livros, teses, dissertações e assim por diante), artigos redundantes de um mesmo autor ou se o foco do trabalho não fosse o uso *Deep learning* para identificar fácies sísmicas. Além desses critérios, caso o artigo não pudesse responder a algumas das questões de pesquisa ele era descartado. Todos os critérios de exclusão são exibidos nas Tabelas 3 e 4.

Tabela 3: Critério de inclusão

#	Critério de inclusão
1	Estudos primários
2	Estudos que façam uso de <i>Deep Learning</i> para suportar identificação de fácies sísmicas
3	Estudos publicados entre Janeiro de 2010 e Janeiro de 2020
4	Estudos que satisfaçam o critério mínimo de qualidade

Fonte: Criada pelo autor.

Tabela 4: Critério de exclusão

#	Critério de exclusão
1	Estudos que não tratem primariamente ou secundariamente de identificação de facies sísmicas sobre dados sísmicos através de <i>Deep learning</i> ;
2	Estudos que realizam o uso de redes neurais convolucionais profundas mas não especifiquem a sua estrutura;
3	Estudos que não demonstrem os resultados;
4	Estudos que não contenham as palavras chaves, no resumo e/ou título e/ou nas palavras-chave do artigo selecionado;
5	Estudos que não contenham respostas à questão de pesquisa;
6	Estudos repetidos, se o mesmo estudo for encontrado em diferentes bases de busca, somente a primeira será considerada;
7	Estudos duplicados, ou seja, trabalhos que apresentem estudos semelhantes, apenas o mais recente ou o mais completo será incluído, a menos que contenham informações complementares;
8	Estudos com texto, conteúdo e resultados incompletos.
9	Estudos que estejam escritos em uma língua diferente do inglês.

Fonte: Criada pelo autor.

3.4.2 Critérios de qualidade

A avaliação da qualidade (QA) dos estudos selecionados é originalmente usada como base para ponderar os dados quantitativos recuperados na meta-análise [38], que é uma estratégia de síntese de dados importante. No entanto, como os dados recuperados nesta revisão foram produzidos por diferentes projetos experimentais, e a quantidade de dados é relativamente pequena, a meta-análise é, portanto, inadequada para tais casos. Para isso não usamos os resultados da avaliação de qualidade para ponderar os dados recuperados.

Os critérios de avaliação da qualidade foram utilizados para orientar a interpretação dos resultados da revisão e para indicar a peso das inferências. Além disso, os resultados da avaliação de qualidade serviram como um critério adicional para a seleção de estudos.

Utilizamos uma série de critérios de qualidade para avaliar o rigor, credibilidade e relevância dos estudos relevantes. Esses critérios são apresentados na Tabela 5. Os critérios CQ1, CQ2, CQ3, CQ4, CQ5, CQ6, CQ7, CQ8, CQ9, CQ10, CQ11 são derivados do trabalho de WEN et al. (2012). Cada pergunta tem apenas três respostas possíveis: *Sim*, *Parcialmente* ou *Não*. A essas três respostas são associados os seguintes pesos: *Sim* = 1, *Parcialmente* = 0,5 e *Não* = 0. Para um determinado estudo, a pontuação de qualidade dele é calculada somando as pontuações de respostas às perguntas de controle de qualidade.

Tabela 5: Critérios de qualidade.

#	Critério de qualidade
CQ1	Os objetivos da pesquisa estão claramente definidos?
CQ2	O contexto do trabalho está descrito de forma adequada?
CQ3	Os métodos de estimativa estão bem definidos e apresentados?
CQ4	O método de avaliação é descrito apropriadamente e justificado?
CQ5	O experimento é aplicado em conjuntos de dados de projeto suficientes?
CQ6	As métricas de avaliação escolhidas estão adequadas para o problema e seus resultados relatados?
CQ7	O método de estimativa proposto é comparado com outros métodos?
CQ8	As conclusões do estudo são claramente declaradas e apoiadas por relatórios ou resultados?
CQ9	As limitações do estudo foram analisadas explicitamente?
CQ10	O conjunto de dados utilizado foi coletado de um ambiente real?

Fonte: Criada pelo autor.

3.4.3 Resultados

A busca de trabalhos nas fontes primárias foi dividida em 5 etapas, onde cada etapa visava realizar a filtragem de trabalhos de acordo com determinados critérios. As fases um a quatro filtraram trabalhos utilizando os critérios de inclusão/exclusão e na fase cinco foram aplicados os critérios de avaliação de qualidade e os trabalhos que não atingissem a nota mínima eram descartados. As etapas aplicadas são exibidas na Tabela 6:

Identificamos 30 artigos relevantes de acordo com o processo de seleção. O número de artigos identificados em cada fase é exibido na Figura 26.

3.5 Extração dos dados

Na fase de extração de dados, analisamos os trabalhos selecionados para recolher informações que pudessem responder as questões de pesquisa da Seção 3.2. As informações foram coletadas e organizadas de acordo com o formulário apresentado na Tabela 7, o qual foi proposto por WEN et al. (2012) e adaptado para o uso desta revisão. Para facilitar o rastreamento dos dados extraídos, rotulamos explicitamente cada estudo com os identificadores das questões de pesquisa correspondentes.

Tabela 6: Passos da seleção de trabalhos

#	Condição para execução do próximo passo	
1	Leitura do título e resumo dos trabalhos.	Se o título ou o resumo não indicassem claramente se os critérios de inclusão/exclusão fossem atendidos, o próximo passo seria executado;
2	Leitura dos resultados.	Se os resultados não indicassem claramente se os critérios de inclusão/exclusão fossem atendidos, o próximo passo seria executado;
3	Leitura da introdução.	Se a introdução não indicasse claramente se os critérios de inclusão/exclusão fossem atendidos, o próximo passo seria executado;
4	Leitura de todo o texto do artigo.	Se o artigo estivesse desalinhado com os critérios de exclusão o mesmo era rejeitado senão, o próximo passo seria executado.
5	Verificação dos critérios de qualidade (Seção 3.4.2) e pontuação obtida.	Se o artigo atingisse uma pontuação abaixo de 5, era descartado.

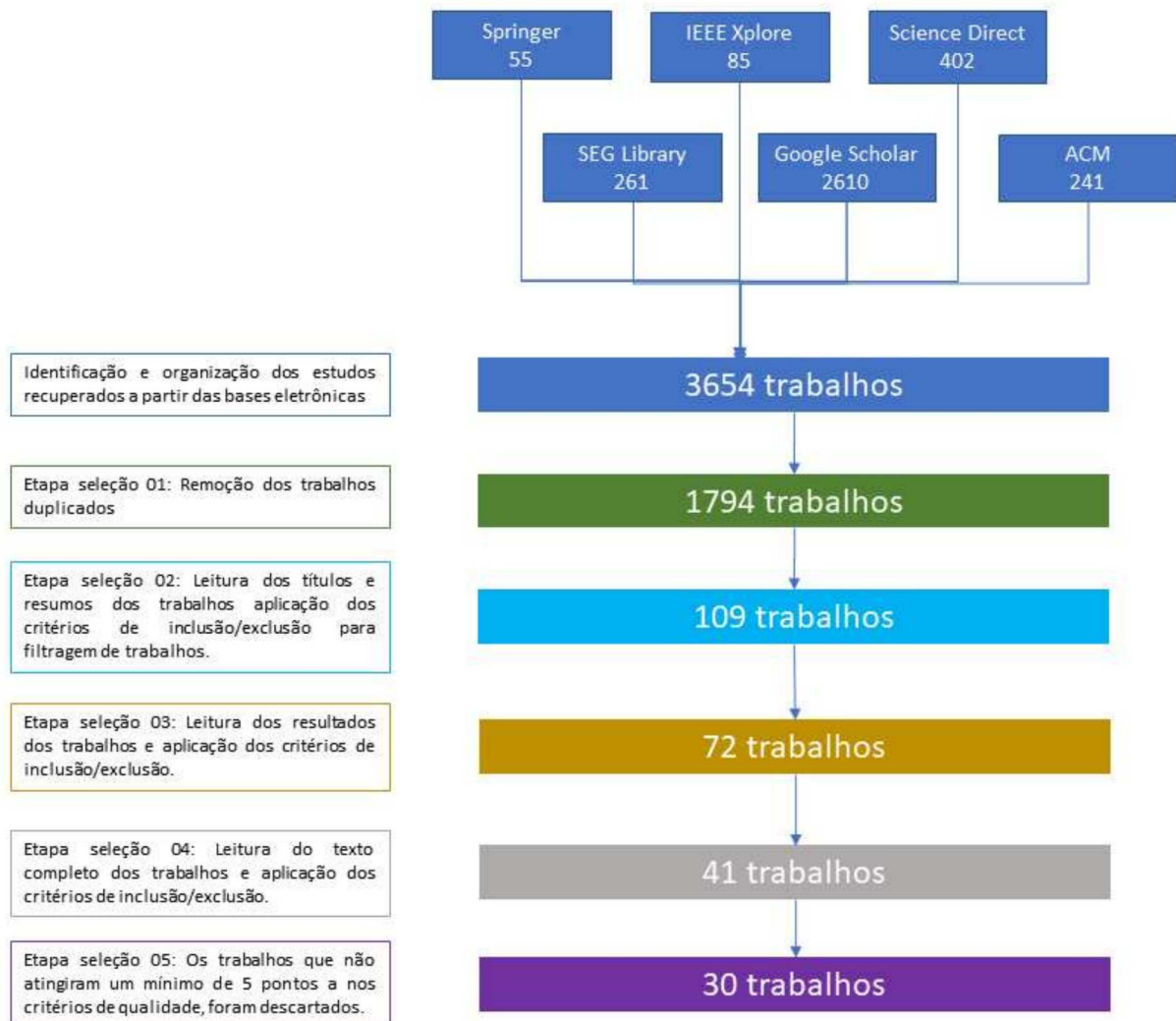
Fonte: Criada pelo autor.

3.6 Síntese dos dados

Nesta fase o objetivo foi agregar evidências dos estudos selecionados para responder às questões de pesquisa. Os dados extraídos nesta revisão incluem dados quantitativos (ex.: valores de precisão de estimativa) e dados qualitativos (ex.: pontos fortes e fracos das técnicas de ML). Foram empregadas as seguintes estratégias para sintetizar os dados extraídos pertencentes às questões de pesquisa desta revisão. As estratégias de síntese são explicadas em detalhes a seguir:

- Para os dados relativos às questões de pesquisa QP1 e QP2, utilizamos o método de síntese narrativa. Ou seja, os dados foram tabulados de acordo com as questões. Algumas ferramentas de visualização, incluindo gráfico de barras e pizza foram usadas para aprimorar a apresentação da distribuição das técnicas de *Deep Learning*.
- Para os dados relativos às questões de pesquisa QP3 e QP4, que se concentram nas métricas e esquemas de representação de dados de entrada das redes neurais, usamos o método de contagem de votos. Ou seja, contamos quantas vezes um determinado item ocorre no conjunto de informações e comparamos essa informação com os outros itens para ter como saída quais os valores de parâmetros e representação de dados mais utilizados nos trabalhos selecionados.

Figura 26: Processo de pesquisa e seleção de artigos



Fonte: Criada pelo autor.

3.7 Resultados

Um total de 30 estudos passaram nos critérios de inclusão/exclusão e qualidade e seus dados foram extraídos. Antes de apresentar os resultados e análises para cada uma das questões de pesquisa, discutimos os resultados da avaliação da qualidade e fornecemos uma visão geral das características gerais dos trabalhos.

3.7.1 Resultados da avaliação de qualidade

A avaliação da qualidade dos estudos selecionados é útil para aumentar a precisão dos resultados da extração de dados. Essa avaliação ajudou a determinar a validade das inferências extraídas e a verificar a credibilidade e síntese dos resultados.

Tabela 7: Formulário de extração de dados

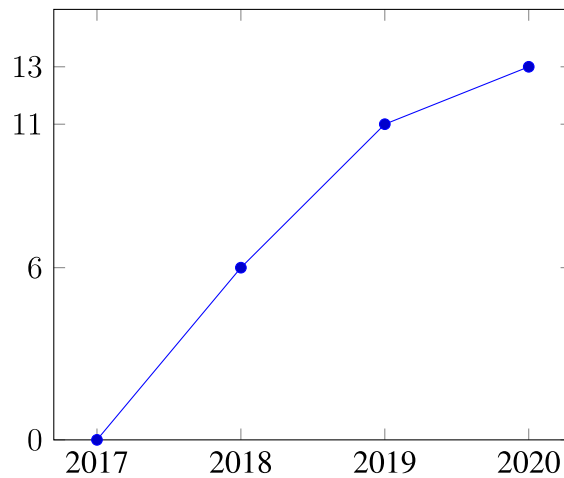
Campos do formulário de extração de dados
Identificador do artigo
Ano de publicação
Nome dos autores
Fonte
Título do artigo
Tipo de estudo (experimento, caso de estudo, pesquisa)
QP1 (Questão de pesquisa)
Técnicas de <i>Deep Learning</i> utilizadas para se identificar fácies sísmicas
QP2 (Questão de pesquisa)
<i>Datasets</i> utilizados nos experimentos
Métodos de validação utilizados nos experimentos
QP3 (Questão de pesquisa)
QP4 (Questão de pesquisa)
Modelos de <i>Deep Learning</i> com os quais foi comparado
QP5 (Questão de pesquisa)
Vantagens da técnica de <i>Deep Learning</i> aplicada em relação a identificação de sismofácies
Desvantagens da técnica de <i>Deep Learning</i> aplicada em relação a identificação de sismofácies

Fonte: Criada pelo autor.

Os resultados da avaliação da qualidade são apresentados na Tabela 8 de acordo com as questões de avaliação descritas na Tabela 5. A pontuação de todos os estudos não é inferior a 5 e a pontuação média é de 7,53. Escolhemos o mínimo de 5 de qualidade com o objetivo de estabelecer um limite de qualidade mínimo para os artigos. Em relação às médias dos critérios de qualidade específicos, CQ1, CQ2, CQ8, CQ10 receberam as pontuações médias mais altas ($\geq 0,9$); CQ3, CQ4, CQ5 E CQ6 receberam, em média, pontuações intermediárias entre 0,65 e 0,77; CQ7 e CQ9 receberam as pontuações médias mais baixas ($<0,5$). O valor baixo de 0,40 para o critério QC7 indica que a maioria dos estudos não se preocupa em comparar os seus resultados com outros modelos de *Deep learning* propostos por outros autores para a tarefa de identificação de fácies sísmicas; e o valor de 0,48 para o critério CQ9 indica que a maioria dos estudos não analisou as suas próprias limitações.

Assim, consideramos que a qualidade geral dos estudos selecionados é aceitável. Pois esses 10 critérios forneceram uma medida da extensão em que poderíamos estar confiantes de que as descobertas de que um determinado estudo poderia dar contribuições valiosas para esta revisão.

Figura 27: Quantidade de trabalhos publicados por ano.



Fonte: Criada pelo autor.

3.7.2 Informações gerais sobre os trabalhos

A seguir, descrevemos as características gerais dos estudos incluídos na revisão: ano de publicação, tipo de fonte, método de pesquisa e forma de aplicação.

3.7.2.1 Ano de publicação

Os artigos revisados foram publicados entre 2018 e 2020. Do ponto de vista temporal (Figura 27), observa-se um número crescente de publicações no âmbito desta revisão desde 2018. O ano de 2020 (46,66%) é o ano com mais publicações, seguido por 2019 (36,67%) e 2018 (16,67%). Com o aparente aumento da quantidade de estudos sobre a aplicação de técnicas de *Deep learning* para a identificação de facies sísmicas, pode-se inferir que há um aumento do interesse de pesquisadores sobre esse tema. Vale ressaltar que, como o processo de busca desta revisão foi finalizado em Janeiro de 2020, pode haver outros artigos sobre esse tema ainda podem não ter sido publicados devido ao tempo de revisão e acertos necessários até a sua versão final ser disponibilizada publicamente.

3.7.2.2 Contexto de aplicação

Os estudos podem ser categorizados em um contexto industrial e acadêmico. A maioria dos artigos (63,33%, 19 estudos, T1, T2, T5, T6, T7, T8, T11, T12, T13, T14, T15, T16, T17, T18, T21, T22, T24, T25, T28) é considerado pertencer a categoria acadêmica-industrial. No entanto, é importante destacar que uma parte dos estudos foi realizada em ambiente puramente industrial (7,5%; 4 estudos, T27, T10, T26, T29), o qual por motivos de propriedade intelectual nem sempre podem fornecer os dados sobre os quais aplicaram os métodos de *Deep Learning*. O

restante dos trabalhos (20%, 6 estudos, T3, T9, T19, T20, T23, T30) foi realizado em ambiente acadêmico. Esses valores demonstram que as aplicações de *Deep Learning* para a identificação de fácies sísmicas despertam o interesse da indústria e do meio acadêmico.

3.7.2.3 Tipo de fonte

Os estudos incluídos nesta revisão podem ser de publicações em periódicos, conferências, *workshops* ou capítulos de livros. A maioria dos estudos são publicações em periódicos (60%; 18 estudos), artigos de conferências (36,66%; 11 estudos), seguidos por publicações em capítulos de livros (3,33%; 1 estudo).

Os 30 estudos selecionados estão distribuídos em 20 fontes de publicação, sugerindo que o uso de técnicas de *Deep Learning* para identificação de fácies sísmicas tem sido uma preocupação generalizada na comunidade de pesquisa. Os principais locais neste tópico de estudo são: AAPG Annual and Exhibition, arXiv, EAGE Conference and Exhibition, Geophysical Journal International, Geophysical Prospecting, GEOPHYSICS, IEEE Dataport, IEEE Geoscience and Remote Sensing Letters, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on Image Processing (ICIP), IEEE Transactions on Geoscience and Remote Sensing, International Joint Conference on Neural Networks (IJCNN), Interpretation, ISPRS International Journal of Geo-Information, Journal of Applied Geophysics, Journal of Geophysics and Engineering, International Conference on Information System and Data Mining, SEG Technical Program Expanded Abstracts, SPE Annual Technical Conference and Exhibition. Esses locais indicam a presença de fontes de áreas de Geociências, Computação visual e inteligência artificial.

3.7.3 RQ1 - Abordagens de *Deep learning*

O objetivo dessa questão foi identificar as principais abordagens de *Deep learning* utilizadas para identificação de fácies sísmicas e se as suas aplicações poderiam trazer benefícios aos trabalhos realizados pelos intérpretes de dados sísmicos.

As abordagens utilizadas podem ser enquadradas em dois grupos: classificação e segmentação. A classificação é responsável por identificar a qual classe uma amostra de entrada da rede neural pertence. Por exemplo, se uma imagem 2D for utilizada como entrada para o treinamento da rede neural e possuir uma dimensão 32x32 (altura x largura) ou seja, 1024 pixels, a rede será treinada para identificar a qual classe todos os 1024 pixels representam. Já a abordagem de segmentação é utilizada para identificar e localizar um objeto no dado sísmico. Por exemplo, nessa abordagem a cada pixel da imagem 2D seria associada uma classe. Com isso, cada objeto geológico existente no dado sísmico poderia ser identificado por uma única classe e assim localizado. Essas duas abordagens baseadas em redes neurais, depois de serem treinadas, obtêm a capacidade de reconhecer as relações não-lineares presentes no dado sísmico e assim identificar

as fácies sísmicas de forma automática. Com isso, o intérprete não necessitaria analisar cada um dos planos que compõem o volume sísmico e guiaria a rede neural para que se especializasse na busca de fácies sísmicas que fossem de seu interesse.

A abordagem predominante a partir da revisão dos trabalhos foi a de segmentação com 73,33% (22 trabalhos: T3-T7, T9, T11, T13-T16, T18, T19, T21, T23, T24, T26, T27, T30) e a classificação correspondeu a 36,66% (11 trabalhos: T2, T8, T10, T12, T17, T20, T22, T25). Sendo que 10% (3 trabalhos: T1, T28, T29) utilizou ambas as abordagens, onde 1 (T29) trabalho comparou as duas abordagens utilizando diferentes estruturas de redes neurais, outro (T1) comparou diferentes tipos de entradas (*patches* ou segmentos, ver Seção 3.7.6 para mais detalhes) para o treinamento de duas redes, e o terceiro (T28) apresentou uma técnica para realizar a rotulação fraca de amostras a serem utilizadas para o treinamento de redes neurais.

O treinamento das redes neurais foi realizado de modo supervisionado em 83% (25 trabalhos: T1, T2, T3, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T17, T19, T20, T21, T22, T23, T24, T25, T27, T29, T30), não-supervisionado em 3% (1 trabalho: T4), e 13% (4 trabalhos: 16, 18, 26, 28) fizeram o uso da abordagem supervisionada e semi-supervisionada. Sendo que a abordagem supervisionada foi a que forneceu previsões mais próximas com os dados interpretados pelo intérprete. Isso se deveu ao fato de que o treinamento supervisionado guia a rede neural em busca das propriedades latentes presentes no dado sísmico que permitem a rede reconhecer e/ou localizar os eventos geológicos de interesse do intérprete. Já nos trabalhos que utilizam o treinamento não supervisionado, a rede era responsável por detectar as propriedades dos dados latentes sem nenhuma ajuda externa, o problema dessa abordagem é que o resultado pode não ter significado geológico como foi discutido no trabalho (T5) (LI et al., 2021). Para o treinamento semi-supervisionado foram utilizados poucos dados rotulados (1 ou 2) onde através de similaridade de textura novas amostras foram extraídas e rotuladas. Essa abordagem tem como vantagem produzir uma grande quantidade de dados para treinamento da rede, mesmo que estes ainda apresentem baixa qualidade, ou seja, muitas amostras não tinham todo o seu conteúdo rotulado (como no trabalho T28 (CIVITARESE et al., 2019)) ou mesmo poderiam ser classificadas erroneamente.

Com as abordagens demonstradas pelos trabalhos revisados utilizando-se redes neurais para classificação/segmentação treinadas de forma supervisionada, semi-supervisionada ou não-supervisionada, foi observado que é factível o uso de *Deep Learning* para assistir o trabalho do interprete, já que a rede neural poderia ser inserida no fluxo diário de seu trabalho e fazer com que não fosse necessário a aplicação de atributos sísmicos para o realce de eventos geológicos de interesse do interprete. O que segundo trabalhos de (DI; GAO; ALREGIB, 2019b; DI; ALREGIB, 2020; CHEVITARESE et al., 2018a) é um processo de tentativa e erro, onde o interprete precisa aplicar cada atributo manualmente e deve conhecer em quais situações um atributo deve ser aplicado ou não. Assim, a rede neural depois de treinada poderia vir a substituir um atributo sísmico.

3.7.4 RQ2 - Algoritmos e arquiteturas utilizadas

O objetivo desta questão de pesquisa foi identificar quais técnicas, algoritmos ou arquiteturas de redes neurais foram testados em dados sísmicos para se identificar fácies sísmicas. A categorização das técnicas, algoritmos e arquiteturas foi feita após a extração dos dados dos estudos, ou seja, durante a extração, foram capturados dados em texto simples dos estudos. A seguir, na etapa de síntese desta revisão, as categorias apresentadas nas Tabelas 9, 10 e 11, foram definidas de acordo com a sua utilização nos estudos. Um detalhe importante nos dados dessas tabelas é que muitos trabalhos não descreviam completamente o método utilizado para o treinamento das redes, o que gerou uma tabela onde nem todas as técnicas puderam ser extraídas e relacionadas com os trabalhos.

Pelas Tabelas 9, 10 e 11, podemos ver quais as técnicas, algoritmos e arquiteturas utilizadas. Observe que um mesmo estudo pode fazer uso de múltiplos itens ao mesmo tempo. Conforme mostrado nas nessas tabelas as redes neurais convolucionais (CNN) são as utilizadas para se identificar fácies sísmicas. Estas redes podem ser agrupadas em dois grupos: com uma camada totalmente conectada ao final ou utilizando o padrão *encoder-decoder*. As redes totalmente conectadas possuem uma camada ao final responsável pela predição da probabilidade de uma amostra de entrada pertencer a uma classe, na maior parte dos trabalhos foi utilizada a função de ativação *softmax* para isso, como visto na Tabela 9. A arquitetura *encoder-decoder* de rede é a responsável pela segmentação dos objetos geológicos, onde os *pixels* da imagem de saída são cada um classificado sendo pertencente a uma classe. Dos trabalhos analisados, 46,67% (14 estudos) fizeram o uso de redes neurais classificação (totalmente conectadas), 43,33% (13 estudos) utilizaram o padrão *encoder-decoder*. No ano de 2018, 2 trabalhos utilizaram redes convolucionais classificadoras de pequenos segmentos, em 2019, foram 3 e em 2020 mais 3. Já para as redes convolucionais no padrão *encoder-decoder*, em 2018 foram encontrados 3 trabalhos publicados em 2018, 7 em 2019 e 3 em 2020. Esses números demonstram que o padrão de rede mais utilizado é de *encoder-decoder* para redes convolucionais para a segmentação de fácies sísmicas.

Para o treinamento das redes neurais, os trabalhos utilizaram a forma supervisionada e semi-supervisionada. Na supervisionada as amostras eram totalmente rotuladas, já para a semi-supervisionada, somente poucas amostras eram rotuladas (1 a 2) e foram utilizadas como amostras fonte de um algoritmo baseado em similaridade de textura através da técnica de *matrix-factorization* (HONG et al., 2016) para a criação de novas amostras rotuladas fracamente. Isso foi feito por causa da característica de que dados sísmicos rotulados são difíceis de serem obtidos, pois dependem que um intérprete especializado em interpretações sísmicas e de muito tempo para serem feitos (CHEVITARESE et al., 2018a).

Para a fase de treinamento das redes é necessário a escolha de uma função de perda e um otimizador. Assim, pelos dados da Tabela 10 a função *cross-entropy* (BOER et al., 2005) foi a mais utilizada, ou seja, 40% (12 artigos) dos trabalhos a utilizaram. Já as funções *Focal Loss*

(LIN et al., 2020) e *Lovász Loss* (BERMAN; RANNEN; BLASCHKO, 2018) corresponderam a 10% (3 trabalhos) cada uma. Porém, o emprego da função *cross-entropy* sozinha não contempla a necessidade de se considerar as relações espaciais entre os resultados preditos pela rede e os reais. Já que a *cross-Entropy* somente se preocupa com a classificação dos *pixels* em uma determinada classe. Funções como *Coral Loss* (SUN; SAENKO, 2016), *Dice Loss* (FIDON et al., 2017), *Focal Loss*, *Lovász Loss*, *Weak focal loss* (ALAUDAH; SOLIMAN; ALREGIB, 2019) levam em consideração as relações espaciais entre os valores contidos nas amostras. Ou seja, essas funções se mostrariam mais adequadas para o problema de segmentação de objetos sísmicos. Em alguns trabalhos (T10, T9, T18, T30) foi feita a combinação de funções de perda, onde na primeira parte do treinamento era utilizada a função *cross-entropy* e na segunda uma outra função como a *Focal Loss* ou a *Lovász Loss* para realizar o *fine-tune* da rede. A *cross-entropy* foi escolhida para a fase inicial pois a sua execução é mais rápida que as outras citadas segundo os autores dos trabalhos que as utilizaram. Essa combinação, sugere que os pesquisadores estão atentos que para treinar redes neurais convolucionais para identificar fácies sísmicas talvez seja necessário que as funções de perda também levem em consideração as relações espaciais entre os *pixels* das imagens preditas e reais. O que acarreta um interesse em propor funções que combinem as já existentes ou mesmo que possam levar em consideração conhecimentos do domínio de sísmica para calcular o valor de perda.

O que se pode notar é que a maioria dos trabalhos utiliza técnicas parecidas tanto para o treinamento da rede como para a construção de sua arquitetura. As funções de ativação, camadas de *pooling*, otimizadores, as técnicas de *data augmentation* e outras que permitem a rede neural seja treinada em menos tempo ou que normalizem os valores dos pesos a fim de evitar problemas de gradiente devido ao algoritmo de *backpropagation*. Para a função de ativação mais utilizada entre as camadas convolucionais temos a ReLU (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), a qual corresponde a 46,66% (14 trabalhos) analisados, ou seja, há uma hegemonia do uso dessa função em relação à função LeakReLU (GLOROT; BORDES; BENGIO, 2011) que poderia ser utilizadas no lugar da função ReLU. Para a camada de *pooling* presente entre camadas convolucionais da rede, a mais utilizada foi a de *Max-pooling*, o que demonstra que técnicas como *Atrous spatial pyramid pooling* (CHEN et al., 2018), *Global Average pooling*, *Global Pooling* foram pouco exploradas. O otimizador mais utilizado foi o Adam (KINGMA; BA, 2014), o que correspondeu a 43,33% (13 artigos) dos trabalhos, isso expõe que grande parte dos artigos não realizou testes com variados otimizadores para o treinamento da rede neural. As técnicas de *data augmentation* mais utilizadas foram a *Horizontal flipping* em 33,33% (10 trabalhos) dos artigos, *Rotation* em 20% (6 artigos) dos trabalhos, *Vertical flipping* 16,67% (5 artigos) e *White noise* em 13,33% (4 artigos) dos trabalhos. As técnicas como *Blurring*, *Intensity scalling*, *Stretching*, *Scalling*, *Tilting* foram sub-utilizadas. A aplicação de técnicas de *data agumentation* deve ser feita tomando precauções pois a ordem e a posição que as estruturas das fácies sísmicas estão presentes no dado sísmico possuem significado do ponto de vista geológico, e a sua mudança aleatória poderia deturbar este aspecto, o que poderia ser feito através do uso de

Vertical flipping e *Rotation*. Sendo para esta última os ângulos utilizados para a rotação variaram de 10° a 15° nos trabalhos T18 (MILOSAVLJEVIĆ, 2020), T24 (NASIM et al., 2020). Já outros trabalhos ou não informam qual o ângulo utilizado ou realizam uma rotação de 90°, 180°, 270°, como o trabalho T11 (DI; GAO; ALREGIB, 2019b). Já as técnicas que auxiliam o treinamento temos o *Dropout* (SRIVASTAVA et al., 2014) sendo utilizado em 50% (15 artigos) dos trabalhos e *Batch normalization* (IOFFE; SZEGEDY, 2015) com 36,67% (11 artigos) dos trabalhos. Já técnicas como *transfer learning*, *Earling stopping*, *Hypercolumns*, *Model checkpoint*, *Reduce Learning Rate on Plateau*, *Skip-connections* (SRIVASTAVA; GREFF; SCHMIDHUBER, 2015), *Sparcy Auto-encoder* (XU; LU; WANG, 2019), *Spectral attention* (LI et al., 2021), residual blocks (HE et al., 2016) e weight decay foram utilizadas em no máximo 10% dos trabalhos. O uso de *skip-connections* no trabalho T16 (ALFARRAJ, 2019) demonstrou que o tempo de treinamento de uma rede *encoder-decoder* teve seu tempo reduzido de 19 horas para 5 horas, o que sugere que é uma técnica com a capacidade de diminuir o tempo de treinamento sem necessitar de grandes alterações da estrutura da rede neural.

Ao final, podemos notar que as técnicas e estruturas de rede neural utilizadas para a identificação de fácies sísmicas tendem a ter uma hegemonia. O que sugere que há muitas combinações de técnicas a serem exploradas e que a aplicação de redes neurais para essa tarefa ainda se encontra em seus estágios iniciais, havendo uma variação muito pequena (de cerca 10%) de trabalhos que tentam explorar novas alternativas na tentativa de aprimorar os resultados atingidos.

3.7.5 RQ3 - Métricas

Para mensurar os resultados atingidos por um trabalho que faça uso de *Deep Learning* é essencial que métricas de avaliação sejam utilizadas. Essas técnicas possuem como objetivo avaliar a qualidade das predições das redes neurais propostas. É importante que seja utilizada mais de uma métrica já que uma rede neural pode ter uma boa performance quando se utiliza uma métrica, mas seu desempenho pode ser ruim quando uma outra é utilizada. O uso de métricas de avaliação é essencial para que a rede ofereça predições adequadas ao problema no qual se tenta resolver.

A Tabela 12 exibe as métricas identificadas nos trabalhos analisados. Note que um mesmo trabalho pode fazer o uso de diversas métricas ao mesmo tempo. Podemos colocar essas métricas em três grupos. O primeiro seriam métricas para classificação, o segundo, métricas para análise espacial dos resultados e o terceiro de análise estatísticas. Pela Tabela 12 é demonstrado que a maioria dos trabalhos se concentra no uso de métricas de classificação: *Accuracy, Average accuracy, False-positive rate, F1-score, Recall, Precision, Average precision, Class accuracy, Mean Class Accuracy, Mean absolute error*. Sendo estas métricas utilizadas em 17 artigos (T1, T7, T8, T10, T11, T12, T15, T16, T17, T19, T20, T21, T22, T24, T25, T26, T28), o que corresponde a 56,66% dos trabalhos analisados. Já as métricas *Pixel accuracy, Average pixel accuracy, Frequency-weighted intersection over union, IoU, Mean IoU* para análise espacial dos resultados, foram utilizadas em 15 artigos, correspondendo a 50% dos trabalhos. O último corresponde a métricas para análise estatísticas dos resultados: *Pearson coefficient, P-value, Receiver Operating Characteristics, Retrieval Accuracy*; e corresponderam a 13,33% (4 artigos) dos trabalhos.

O que se pode notar pelos valores de porcentagem do parágrafo anterior são métricas voltadas a classificação são as mais utilizadas em trabalhos que envolvem segmentação de objetos. Isso denota que os resultados atingidos por esses trabalhos apresentariam resultados mais adequados à tarefa de segmentação se utilizassem métricas voltadas a isso, o que não ocorreu na maioria dos trabalhos analisados. O uso adequado de métricas possibilita que se crie um conjunto de dados para avaliar a performance das diferentes estruturas e modos de treinamento de redes neurais para a identificação de fácies sísmica, sendo que atualmente, não há um *dataset* de referência para tal tarefa.

Por fim, alguns trabalhos não informaram quais métricas foram utilizadas, o que correspondeu a 16,67% (5 artigos) dos trabalhos; o que os autores fizeram foi a análise e discussão dos resultados baseando-se visualmente nas predições das redes neurais propostas em seus artigos.

3.7.6 RQ4 - Representação de dados sísmicos

O propósito dessa questão de pesquisa foi de verificar como os dados de fácies sísmicas foram utilizados pelos trabalhos analisados. Como as fácies sísmicas compreendem dados

categoricos, há muitas maneiras de transformar essas informações em dados numéricos exigidos pelos algoritmos de *Deep Learning*. Portanto, esse foi o interesse primário ao se analisar como essa transformação pode contribuir para a melhoria dos resultados e quais esquemas de codificação são amplamente utilizados neste tipo de tarefa.

Ao analisar os trabalhos, verificamos que 90% (27 artigos: T1, T2, T3, T4, T5, T7, T8, T9, T10, T11, T13, T18, T19, T20, T21, T23, T24, T25, T27, T28, T29, T30) dos trabalhos utilizaram como entrada dados em duas dimensões (2D) e 10% (10 artigos: T12, T22, T26) em três dimensões (3D). A maioria dos trabalhos utilizou dados 2D devido às limitações de hardware (quantidade RAM de GPU). Sendo o formato 2D requer redes neurais com um menor número de parâmetros treináveis, o que diminui o tempo de treinamento quando se compara uma mesma estrutura de rede, mas com uma entrada e predições em volumes.

As entradas em duas dimensões eram organizadas das seguintes formas: *patches* ou seções sísmicas inteiras. Foram utilizados dois tipos de *patches*. O primeiro sendo fatias de imagens sísmicas 2D, já o segundo de fatias de dados sísmicos de seções sísmica, ou seja, extraídos diretamente do volume sísmico sem serem processados para visualização como nas imagens sísmicas em 2D. Quando os dados sísmicos eram utilizados diretamente em forma de *patches*, esses poderiam representar um conjunto de valores de amplitude sísmica qualquer como entrada da rede neural. Uma segunda forma de representação foi utilizada no trabalho T5 (LI et al., 2021), onde os *patches* eram compostos cada um por somente espectros de frequência que foram associados a um determinado tipo de fácies sísmica de acordo com um intérprete. Uma outra forma da criação dos *patches* era com imagens sísmicas criadas a partir de dados de seções sísmicas; com isso os *patches* eram extraídos de imagens 2D ou 3D para treinamento da rede neural. No caso de volumes sísmicos ele pode ser entendido como sendo composto por planos equidistantes entre si, no qual cada um dos planos pode ser utilizado para a extração dos *patches* como se fosse uma sísmica 2D. O uso de *patches* de imagens sísmicas foi feito em 80% (24 trabalhos: T1, T2, T3, T4, T6, T8, T9, T10, T11, T13, T15, T16, T17, T18, T19, T20, T21, T23, T24, T25, T27, T28, T29, T30), *patches* de dados sísmicos foi feito em 6% (2 trabalhos: T5, T7), o de sub-volumes do volume sísmico em 13% (4 trabalhos: T12, T14, T22, T26) dos trabalhos. O que se observa é uma hegemonia do uso de *patches* de imagens sísmicas e a utilização de sub-volumes ou *patches* de dados sísmicos ainda carecendo de mais estudos para se verificar se redes treinadas com eles teriam predições com maior fidelidade ou se o tempo de treinamento seria reduzido.

3.7.7 RQ5 - Comparação entre modelos

O propósito dessa questão de pesquisa foi de verificar se há algum modo padrão para se avaliar as técnicas de Deep Learning utilizadas para a identificação de fácies sísmicas. Através desses modelos poderíamos saber quais os *datasets* mais utilizados, qual a forma de comparação de modelos de redes amplamente utilizada.

A partir da extração dos dados dos trabalhos, foi possível identificar que 11 *datasets* foram utilizados para o treinamento das redes neurais. O *dataset* mais utilizado foi o *Netherlands F3 block* em 36,67% (11 artigos) dos trabalhos. Somente em dois trabalhos (T14, T20) não pudemos identificar qual *dataset* foi utilizado. Quais *datasets* foram utilizados e os seus trabalhos associados são apresentados na Tabela 13. Dentro os *datasets* utilizados que estão disponíveis para download e possuem amostras rotuladas temos os seguintes: *Facies-Mark*(ALFARRAJ, 2019), *LANDMASS 1 e 2 dataset*(ALAUDAH; ALREGIB, 2015), *MalenoV interpretation*, *Penobscot dataset*(BARONI et al., 2019), *StData-12*(DI; GAO; ALREGIB, 2019b), *TGS Salt Identification Challenge*(KAINKARYAM et al., 2019). Os *datasets* *Facies-MARK*, *LANDMASS 2 dataset*, *MalenoV interpretation*, *Penobscot dataset*, *StData-12* foram utilizados nos trabalhos analisados para a identificação de múltiplas fácies sísmicas. Nesses *datasets* todos os valores que compõem cada uma das amostras de treinamento que correspondiam a alguma fácies sísmica eram associados com um determinado rótulo. Já os *datasets* *LANDMASS 1 dataset* e *TGS Salt Identification Challenge* foram utilizados para se realizar a identificação de domos de sal. Desses *datasets* somente o *StData-12* e *Facies-Mark* são adequados para o uso de redes com a arquitetura *encoder-decoder* pois as máscaras contidas neles possuem todos os *pixels* rotulados. Cabe se salientar que a maioria dos trabalhos (18 de 30 artigos) utilizou dados com interpretações que não estão disponíveis publicamente, o que torna a análise dos dados de entrada para treinamento da rede impossível. A partir dessas informações se pode chegar à conclusão que não há um *dataset* totalmente rotulado que seja amplamente utilizado para o treinamento de redes neurais. Porém, com o aumento do interesse de aplicação de redes neurais sobre dados sísmicos os *datasets* rotulados listados nesse parágrafo podem passar a ser utilizados em mais trabalhos futuros e ser uma fonte de referência para análise da performance deles.

Entre os estudos analisados, 70% (21 trabalhos: T1, T2, T3, T6, T9, T10, T11, T12, T13, T14, T15, T16, T18, T20, T21, T22, T23, T25, T27, T28, T30) não compararam a predições de suas redes com as de outros modelos de redes neurais, ou seja, somente 30% (9 trabalhos: T4, T5, T7, T8, T17, T24, T26, T29, T19) fizeram essa comparação. O modo de avaliação dos artigos que não fizeram a comparação foi baseado na narração do autor sobre os resultados preditos pela rede neural. Nos trabalhos T4 (XU; LU; WANG, 2019), T5 (LI et al., 2021), os autores comparou a predição da sua rede proposta com outros modelos de rede que se propuseram a fazer a segmentação de domos de sal. A métrica utilizada foi baseado no índice IoU, onde era calculado um valor de IoU para cada *patch* de imagem predito pela rede neural e depois a média era calculada sobre eles. O trabalho T4 comparou as suas predições com as redes FCN (LONG; SHELHAMER; DARRELL, 2015), PSPNet (ZHAO et al., 2017), DeepLab (CHEN et al., 2017), EncNet (ZHANG et al., 2018), DenseASPP (YANG et al., 2018), OCNNet (YUAN; WANG, 2018), DANet (FU et al., 2019). O trabalho T5 comparou as suas predições com as das redes FCN16/32 (LEE et al., 2017), SegNet (BADRINARAYANAN; KENDALL; CIPOLLA, 2017), U-Net (RONNEBERGER; FISCHER; BROX, 2015). O trabalho T7 (ZHANG et al., 2021) calculou a média dos índices IoU a partir das predições da sua rede proposta as da rede neural

Waldeland CNN (WALDELAND; SOLBERG, 2016) e com os atributos sísmicos *Chaos*, *RMS amplitude*, *Variance*. O trabalho T8 (ZENG; JIANG; CHEN, 2019) comparou a acurácia das predições das redes *Waldeland CNN* (WALDELAND; SOLBERG, 2016), *VGG16* (SIMONYAN; ZISSERMAN, 2014) e ResNet (HE et al., 2016). O trabalho T17 (WU et al., 2019a) comparou visualmente as predições de sua rede neural proposta com as da rede FaultSeg3D (WU et al., 2019b). O trabalho T19 (LIU et al., 2020b) comparou os valores de acurácia e os índices de IoU atingidos durante o treinamento da sua rede proposta como os das redes U-Net, LinkNet (CHAURASIA; CULURCIELLO, 2017) e PSPNet (ZHAO et al., 2017). O trabalho T24 (NASIM et al., 2020) criou várias versões da sua rede proposta e as comparou as suas predições utilizando as métricas *accuracy*, *mean class accuracy*, *Frequency-weighted intersection over union*, *Mean IoU*. O trabalho T26 (CHEVITARESE et al., 2018a) comparou as predições das suas duas redes propostas (uma GAN e uma FCN) utilizando tanto a acurácia como uma inspeção visual das predições. O trabalho T29 (ALAUDAH; ALFARRAJ; ALREGIB, 2019) comparou os valores médio dos índices de IoU atingidos durante o treinamento da sua rede proposta como os das redes DanetFCN2/3 (CHEVITARESE et al., 2018b) e FCN8 (LONG; SHELHAMER; DARRELL, 2015). Em 6 trabalhos de 8 que fizeram comparações entre modelos de redes neurais, a métrica IoU que leva em consideração as relações espaciais entre as predições da rede neural e a amostra real foi utilizada. Esses dados expressam que muitos trabalhos não se preocuparam em comparar seus resultados com outros modelos e que há a possibilidade tanto da criação de *benchmarks* para o teste de predições como a definição de métricas padrões para a avaliação das predições entre diferentes modelos.

3.8 Considerações

A segmentação é um processo que resulta na delimitação dos objetos contidos nas imagens. A aplicação de redes neurais sobre dados sísmicos fornece a habilidade de aprender as características essenciais que permitem a construção de uma imagem de saída contendo estruturas geológicas segmentadas. Com redes neurais que utilizam a arquitetura *encoder-decoder* poderíamos simular a forma de trabalho do intérprete em algum nível. Assim, treinar uma rede neural *encoder-decoder* para identificar *pixels* entre geometrias de fácies sísmicas e, extrair características essenciais, para reconstruir uma saída com regiões segmentadas que representam essas fácies separadas por linhas. Ao contrário da tarefa de classificação, a segmentação pode fornecer predições de objetos com mais detalhes e foi esta a escolhida a ser utilizada neste trabalho. A maioria dos trabalhos que usa redes neurais para identificar fácies sísmicas enfoca na segmentação de apenas um tipo de fácies e usa uma função de perda que não considera a distribuição espacial dos *pixels* nas previsões de resultados. Nestes trabalhos também há a utilização de dados sintéticos ou reais tanto em duas ou três dimensões, como foi visto na Seção 3.7.6. A Tabela 14 exhibe um resumo do que foi apresentado neste parágrafo e contém os artigos que possuem uma alta correlação com este trabalho.

Neste trabalho, apresentamos a *Deep neural network for facies Segmentation* (DNFS, para abreviar) na Seção 4.1 do Capítulo 4. A qual é uma rede para segmentação binária de fácies sísmicas. Com isso, um número arbitrário de fácies poderia ser segmentado pois apenas nos focaríamos nas transições entre essas fácies 4.3. Para levar em consideração a relação espacial entre os pixels durante o processo de treinamento da rede, as funções de perda *cross-entropy* e *Jaccard loss* poderiam ser combinadas linearmente. Onde a *cross-entropy* seria responsável por atribuir as classes aos *pixels*. A função *Jaccard loss* seria responsável por mensurar o quanto espacialmente as imagens preditas e as reais estão correlacionadas. Essa combinação é proposta na Seção 4.2.

Tabela 8: Resultados da avaliação de qualidade dos artigos selecionados.

#	Autor	Ano	CQ1	CQ2	CQ3	CQ4	CQ5	CQ6	CQ7	CQ8	CQ9	CQ10	Total
T1	DI; ALREGIB (2020)	2020	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	10,00
T2	GUO et al. (2020)	2020	1,00	1,00	1,00	1,00	1,00	1,00	0,50	1,00	0,00	1,00	8,50
T3	ZHAO; MUKHO-PADHYAY (2018)	2018	1,00	1,00	0,00	0,00	1,00	0,50	0,00	1,00	0,00	1,00	5,50
T4	XU; LU; WANG (2019)	2019	1,00	1,00	0,00	1,00	1,00	0,00	0,00	1,00	0,00	1,00	6,00
T5	LI et al. (2021)	2020	1,00	1,00	1,00	0,50	1,00	1,00	1,00	1,00	0,00	1,00	8,50
T6	WANG; YANG; MA (2018)	2018	1,00	1,00	0,50	0,00	0,50	1,00	0,00	1,00	0,00	0,00	5,00
T7	ZHANG et al. (2021)	2020	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	10,00
T8	ZENG; JIANG; CHEN (2019)	2019	1,00	1,00	0,50	0,50	0,00	1,00	0,00	1,00	1,00	1,00	7,00
T9	GUPTA et al. (2019)	2019	1,00	1,00	0,50	0,50	0,50	1,00	1,00	1,00	1,00	1,00	8,50
T10	DRAMSCH; LüTHJE (2018)	2018	1,00	1,00	1,00	0,50	0,50	1,00	0,00	1,00	1,00	1,00	8,00
T11	DI; GAO; ALREGIB (2019b)	2019	1,00	1,00	0,50	1,00	0,50	1,00	0,00	1,00	1,00	1,00	8,00
T12	GUAZZELLI; ROISENBERG; RODRIGUES (2020)	2020	1,00	1,00	1,00	1,00	1,00	0,50	0,00	1,00	0,00	1,00	7,50
T13	MATTOS et al. (2020)	2020	1,00	1,00	0,50	1,00	1,00	0,50	0,00	1,00	1,00	1,00	8,00
T14	TSCHANNEN et al. (2020)	2020	1,00	1,00	0,50	0,50	0,50	0,50	0,00	1,00	1,00	1,00	7,00
T15	ALAUDAH; SOLIMAN; ALREGIB (2019)	2019	1,00	1,00	0,50	0,50	1,00	1,00	0,00	0,50	0,00	1,00	6,50
T16	ALFARRAJ (2019)	2019	1,00	1,00	1,00	0,50	1,00	1,00	0,00	1,00	1,00	1,00	8,50
T17	WU et al. (2019a)	2019	1,00	1,00	1,00	0,50	1,00	0,50	1,00	1,00	1,00	0,00	8,00
T18	MILOSAVLJEVIĆ (2020)	2020	1,00	1,00	1,00	1,00	1,00	0,50	1,00	1,00	0,00	1,00	8,50
T19	LIU et al. (2020b)	2020	1,00	1,00	0,50	0,50	0,00	0,50	0,00	1,00	0,00	1,00	5,50
T20	ALAUDAH; GAO; ALREGIB (2018)	2018	1,00	1,00	0,50	0,50	0,50	0,00	0,00	1,00	0,00	1,00	5,50
T21	ALFARHAN et al. (2020)	2020	1,00	1,00	0,50	1,00	0,50	1,00	0,50	1,00	0,00	1,00	7,50
T22	MOSSER; OLIVEIRA; STEVENTON (2019)	2019	1,00	1,00	0,50	0,50	0,50	0,50	0,00	0,50	0,50	1,00	6,00
T23	SHI; WU; FOMEL (2019)	2019	1,00	1,00	0,50	1,00	1,00	1,00	0,00	1,00	1,00	1,00	8,50
T24	NASIM et al. (2020)	2020	1,00	1,00	1,00	0,50	1,00	1,00	1,00	0,50	1,00	1,00	9,00
T25	LIU et al. (2020a)	2020	1,00	1,00	0,50	0,50	1,00	0,50	1,00	0,50	0,00	1,00	7,00
T26	CHEVITARESE et al. (2018a)	2018	1,00	1,00	0,50	0,50	1,00	0,50	0,00	0,50	0,50	1,00	6,50
T27	DI; WANG; ALREGIB (2018)	2018	1,00	1,00	0,50	0,50	0,50	0,50	1,00	0,50	0,50	1,00	7,00
T28	CIVITARESE et al. (2019)	2019	1,00	1,00	0,50	0,50	1,00	1,00	1,00	1,00	0,00	1,00	8,00
T29	ALAUDAH; ALFARRAJ; ALREGIB (2019)	2019	1,00	1,00	1,00	0,50	0,50	1,00	1,00	1,00	1,00	1,00	9,00
T30	ISLAM (2020)	2020	1,00	1,00	0,50	1,00	1,00	1,00	0,00	1,00	0,00	1,00	7,50
Média			1,00	1,00	0,65	0,65	0,77	0,75	0,40	0,90	0,48	0,93	7,53

Fonte: Criada pelo autor.

Tabela 9: Sumário das técnicas utilizadas nos trabalhos analisados, parte 1 de 3.

Função de perda	Estudo
Balanced cross-entropy loss	T21
Coral Loss	T3, T24
Cross-categorical loss	T8
cross-entropy loss	T6, T8, T9, T11, T13, T18, T19, T22, T24, T25, T26, T30
Dice loss	T30
Focal Loss	T3, T15, T18
Lovasz Loss	T30, T8, T9
Weak focal loss	T15
Composed loss function	T9, T15, T18, T20,
Tipo de treinamento	Estudo
Supervisionado	T1, T2, T3, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16, T17, T19, T20, T21, T27, T28
Semi-supervisionado	T4, T15, T18, T25, T26, T28
Função de ativação	Estudo
Leaky ReLU	T9, T30
RELU	T1, T7, T8, T10, T12, T14, T16, T17, T18, T22, T23, T26, T27, T29
Softmax	T2, T7, T8, T10, T14, T17, T23, T25, T26
Sigmoid	T10, T19, T22
Função de <i>pooling</i>	Estudo
Atrous spatial pyramid pooling	T7, T24
Depthwise separable convolution	T7
Global Average pooling	T3, T17, T20, T24,
Max pooling	T10, T11, T12, T14, T16, T18, T19, T20, T22, T23, T25
Unpooling	T18, T23
Otimizadores	Estudo
Adam	T5, T7, T8, T9, T10, T11, T13, T14, T19, T20, T22, T24, T26
Adadelta	T17, T24
RMSProp	T29

Fonte: Criada pelo autor.

Tabela 10: Sumário das técnicas utilizadas nos trabalhos analisados, parte 2 de 3.

Arquitetura de rede	Estudo
Danet-FCN (FCNN)	T29
Danet-FCN2 (FCNN)	T13, T29
DeepLabv3+ (ED)	T7
Encoder-decoder CNN	T3, T15, T16, T23, T27
FCN8	T18, T29
FCN16	T29
GAN	T26
LeNet network (FCNN)	T10
ResNet50 (FCNN)	T8
ResNet (FCNN)	T9
Se-ResNet (FCNN)	T30
StNet (ED)	T11
U-Net (ED)	T6, T9, T21, T22, T29, T30
VGG16 (FCNN)	T8, T29
VGGNET	T26
Waldeland CNN (FCNN)	T8
Data agumentation	Estudo
Blurring	T24
Horizontal flipping	T3, T7, T9, T11, T14, T15, T16, T18, T19, T24
Intensity scalling	T19
Rotating	T7, T9, T11, T15, T18, T24
Stretching	T7
Scaling	T9
Tilting	T7, T9
Vertical flipping	T11, T7, T9, T14, T24
White noise	T14, T15, T16, T24
Técnica	Estudo
Batch normalization	T7, T8, T9, T10, T14, T16, T17, T22, T23, T27, T29
Dilated convolution	T5
Dropout	T1, T7, T8, T9, T10, T11, T12, T14, T17, T18, T23, T24, T25, T27, T29
Early stopping	T8, T10, T19
Fine-tune	T8, T21
He kernel initialization	T9
Hypercolumns	T3
Model Checkpoint	T19
Normal distribution initializer	T25
ReduceLROnPlateau	T19, T30
Residual Blocks	T3, T6, T24
Skip-connections	T3, T6, T15
Sparcy Auto-encoder (SAE)	T4
Spectral attention	T5
Transfer-learning	T8, T21, T24

Fonte: Criada pelo autor.

Tabela 11: Sumário das técnicas utilizadas nos trabalhos analisados, parte 3 de 3.

Técnica	Estudo
Transposed Residual Blocks	T24
Up-Sampling (nearest neighbor interpolation)	T14
Xavier kernel initialization	T29
Weight decay	T5, T24, T27
Weakly supervised matrix-factorization	T18, T28

Fonte: Criada pelo autor.

Tabela 12: Métricas de avaliação utilizadas nos trabalhos. Primeiro campo se refere a métrica, o segundo campo, o identificador do trabalho, terceiro campo a quantidade de artigos que utilizaram a métrica. O item com * representa que não foi utilizada uma métrica de avaliação, em seu lugar o autor de trabalho fez uma análise visual dos resultados.

Métrica	Estudo	Quantidade
Accuracy	T10, T11, T12, T22, T25, T26	6
Average accuracy	T1, T7, T15	3
False-positive rate	T1	1
F1-score	T1, T10, T15, T21, T22, T26	6
Recall	T12, T17, T20, T21, T22, T24	6
Precision	T12, T17, T20, T21, T22, T24, T28	7
Average precision	T26, T28	2
Class accuracy	T12, T16, T19, T24	4
Mean Class Accuracy	T12, T15, T16, T17, T24	5
Mean absolute error	T8	1
Pixel accuracy	T12, T15, T16, T24	4
Average pixel accuracy	T22	1
Frequency-weighted intersection over union	T12, T15, T16, T24	4
IoU	T3, T5, T6, T9, T19, T21, T30	7
Mean IoU	T7, T24, T27, T29	4
Pearson coeficiente	T24	1
P-value	T24	1
Receiver Operating Characteristics (ROC)	T17, T20, T28	3
Retrieval Accuracy (PA)	T28	1
Não informada	T2, T4, T14, T18, T23	5

Fonte: Criada pelo autor.

Tabela 13: Datasets utilizados para treinamento das redes neurais

Dataset	Estudo
Dataset sintético	T6, T17
<i>Facies-Mark</i>	T12, T15
<i>Great South Basin</i>	T25
<i>LANDMASS-1 dataset</i>	T21
<i>LANDMASS-2 dataset</i>	T28
<i>MalenoV interpretation</i>	T8
<i>Netherlands F3 block</i>	T4, T5, T7, T10, T13, T16, T18, T21, T23, T24, T26, T28
<i>Penobscot dataset</i>	T10, T27, T29
<i>SEAM</i>	T1, T2, T9, T22
<i>Stanford VI-E synthetic reservoir</i>	T12
<i>StData-12</i>	T11
<i>TGS Salt Identification Challenge</i>	T3, T19, T30
Não informa	T14, T20

Fonte: Criada pelo autor.

Tabela 14: Características resumidas dos trabalhos analisados. Os artigos que compõem a tabela são aqueles que possuem alta correlação com este trabalho.

Artigo	Função de perda	Tarefa	Segmenta várias fácies	Dados reais	Dimensão de entrada
Di (2018)	Cross-entropy	Segmentação	Sim	Sim	2D
Zheng (2019)	Cross-entropy	Classificação	Não	Não	3D
Zhao (2018)	Cross-entropy	Segmentação e classificação	Não	Sim	2D E 3D
Zeng (2019)	Cross-entropy e Lovász-Softmax	Classificação	Não	Sim	2D
Wu (2019)	Cross-entropy	Classificação	Sim	Não	2D e 3D
Di, Haibin (2019)	Cross-entropy	Segmentação e classificação	Não	Não	2D
Lima (2020)	Cross-entropy e Jaccard loss	Segmentação	Sim	Sim	2D

Fonte: Criada pelo autor.

4 MÉTODO

Este trabalho se propõe a utilizar uma rede neural do tipo *encoder-decoder* (Seção 2.2.2) para a segmentação de fácies sísmicas (Seção 2.1) em imagens sísmicas 2D. Duas arquiteturas de redes foram selecionadas como bases: U-Net e StNet. A U-Net foi apresentada por (RONNEBERGER; FISCHER; BROX, 2015) e aplicada sobre dados biomédicos para a segmentação de células cancerígenas, a sua seleção se deve a sua similaridade no tipo de dado a ser segmentado, dados ruidosos e máscaras binárias (somente as cores pretas e brancas). A StNet foi proposta por (DI; GAO; ALREGIB, 2019a) e aplicada sobre dados sísmicos para a segmentação de fácies sísmicas se baseando em cores para cada região segmentada. Baseado na estrutura dessas duas redes foi criada uma terceira rede chamada DNFS (*Deep neural network for facies segmentation*).

A quantidade de filtros em cada uma das camadas da DNFS foi modificada afim de verificar a escalabilidade da DNFS e de se encontrar uma configuração de rede com o menor número de parâmetros capaz de prover previsões referentes as regiões que separam fácies sísmicas. Assim, essa rede teria a possibilidade de ser utilizada em um fluxo de trabalho diário de intérpretes de dados sísmicos, ou seja, a rede poderia ser treinada em pouco tempo e utilizada como um método de auxílio à interpretação de volumes sísmicos. Além da rede DNFS, foi criada uma função de perda composta e uma métrica aplicada somente sobre as previsões do conjunto de dados de testes. A função de perda é baseada na combinação linear entre *Cross-entropy* e *Jaccard Loss* e a métrica foi baseada em *Dice-coefficient* (DICE, 1945) e fornece a porcentagem de *pixels* pretos preditos corretamente pela rede neural.

Para treinar a rede em pouco tempo, pensamos em um modo de minimizar a quantidade de características que deveriam ser extraídas pela rede neural (Seção 2.2.1). Com isso realizamos a adaptação de dois *datasets*, baseado-nos na premissa de que se nos focássemos nas regiões divisórias entre fácies sísmicas poderíamos diminuir a quantidade de parâmetros da rede, já que o treinamento passaria a ser binário. Os *datasets* adaptados são o StData-12 proposto por (DI; GAO; ALREGIB, 2019b) e Facies-Mark proposto por (ALFARRAJ, 2019), onde ambos são utilizados para a segmentação de múltiplas fácies sísmicas. Antes do treinamento da rede, as imagens contidas nesses *datasets* foram pré-processadas para a sua normalização e atribuição de classes aos *pixels* da máscara de saída a ser predita de forma binária. O treinamento das diferentes configurações de filtros para a rede DNFS foi feito com os mesmos hiper-parâmetros, com exceção do valor do *batch*, o qual era diferente de acordo com a quantidade inicial de filtros de cada configuração.

Após o treinamento da rede (Seção 5.1), foram exibidas as imagens das máscaras originais e as máscaras preditas para que fossem comparadas visualmente (Seção 5.3). A verificação quantitativa da previsão foi feita por uma métrica baseada em *Dice-coefficient* (DICE, 1945). Ao contrário do *Dice-coefficient*, esta métrica somente levou em consideração os *pixels* pretos no cálculo do índice, ou seja, *pixels* brancos foram ignorados por ela já que o foco deste trabalho está na previsão das linhas pretas divisórias entre as fácies sísmicas. O uso dessas linhas permitiu

a utilização de classificação binária de para cada pixel sobre um fundo em cor branca.

4.1 Arquitetura

Esta seção apresenta a *Deep neural network for facies segmentation* (DNFS), uma variante de StNet (DI; GAO; ALREGIB, 2019a) e U-Net (RONNEBERGER; FISCHER; BROX, 2015) otimizada para segmentação de fácies sísmicas. A DNFS é um rede *encoder-decoder* que utiliza uma camada de conexão entre as partes *encoder* e *decoder* e camadas de convolução transposta na parte *decoder*.

Usamos um tipo de segmentação denominado segmentação semântica, em que a rede neural extrai características dos dados para reconhecer a localização dos objetos em uma imagem, não sendo feita distinção entre objetos da mesma classe (LIANG et al., 2018). A utilização da arquitetura *encoder-decoder* permite extrair as características latentes da imagem de entrada para reconstruir uma saída com base nas latentes. As que foram extraídas da imagem de entrada são as transições entre fácies sísmicas. Por isso, usamos apenas duas classes: 0 e 1. A classe 0 (zero) compreende um local sem transição entre as fácies sísmicas e a classe 1 (um) representa um local de transição. Para realizar essa tarefa com segmentação binária (onde existem apenas duas classes), criamos a DNFS.

Das três arquiteturas de redes utilizadas neste trabalho, as da StNet (Fig. 29) e U-Net (Fig. 30) tiveram as suas estruturas de camadas e filtros propostas em seus trabalhos mantidas. Já para a rede DNFS, a quantidade de filtros foi proporcional aos seguintes fatores: 4, 8, 16, 32, 64. A escolha desses fatores foi devida ao fato de que utilizamos a mesma *GPU* para treinar os modelos e fatores maiores que 64 geravam estouro de memória durante o processamento por não haver memória RAM suficiente. Essas modificações foram feitas para se averiguar qual o número mínimo de parâmetros da rede DNFS com a qual se obteria predições das linhas divisórias entre as fácies sísmicas sem haver a modificação da estrutura base da rede. Para todas as redes, após cada camada convolucional foram utilizadas as camadas *ReLU* e *BatchNormalization*, ao final de cada bloco de convolução (conjunto de camadas com o mesmo número de filtros) e às subjacente camadas *UpSampling* foi utilizada a técnica de *Dropout*. A estrutura da DNFS é composta por características presentes tanto na StNet como na U-Net. Na parte *encoder* foram utilizadas camadas convolucionais empilhadas em blocos, onde cada camada possuía o mesmo número de filtros. Na parte *decoder*, foram utilizadas camadas de convolução transposta. Entre os blocos que formam as partes *encoder* e *decoder* com mesmo número de filtros foi utilizada a técnica chamada *skip-connections* (SRIVASTAVA; GREFF; SCHMIDHUBER, 2015), a qual tem o objetivo de evitar o problema de *Vanishing gradient problem* (HOCHREITER, 1998) onde os valores de atualizações calculados se tornam próximos a zero ou zero e o algoritmo de aprendizagem da rede não atualiza seus pesos.

Para encontrar a menor variação da DNFS, variamos o número de filtros de acordo com o exibido na Figura 28 onde a variável Ω era substituída por um valor dos fatores. Na parte *decoder*

foram utilizados blocos compostos por camadas de convolução transposta (WANI et al., 2019), as quais são usadas para aumentar a amostragem do mapa de características de entrada para um mapa de características de saída com as mesmas dimensões. A diferença entre as camadas de convolução e convolução transposta na arquitetura *encoder-decoder* é que a primeira realiza a compressão dos dados para uma representação latente e a segunda efetua a descompressão dessas características. No caso deste trabalho, essa descompressão foi guiada para prover características que representem as linhas separatórias entre as fácies sísmicas.

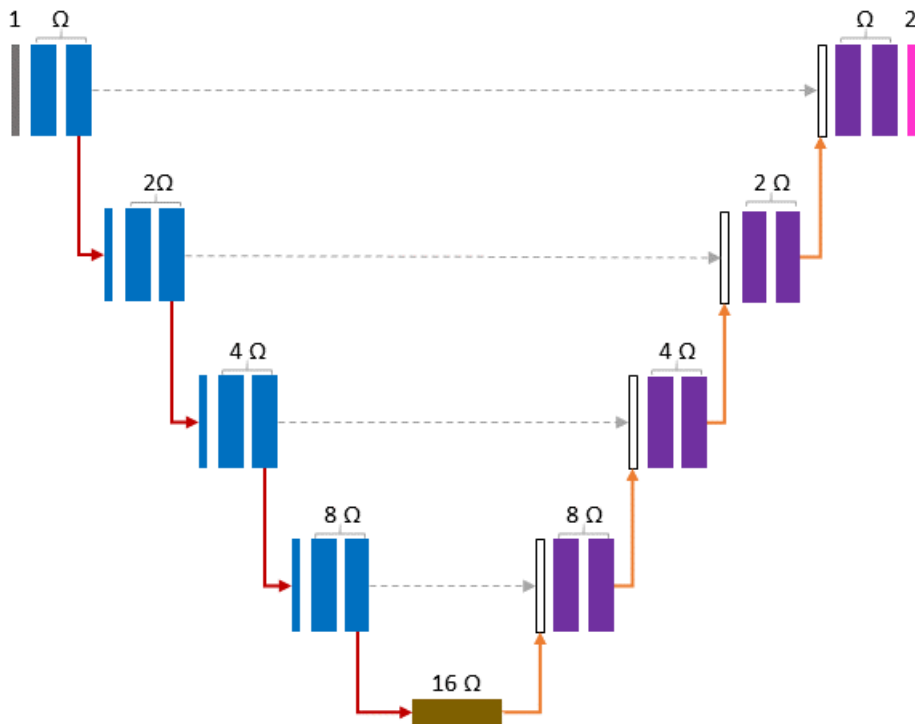
Para otimizar a quantidade de parâmetros da DNFS e ao mesmo tempo permitir a segmentação de múltiplas fácies sísmicas simultaneamente, a DNFS foi criada tendo como foco a segmentação binária. Onde as imagens preditas por ela deveriam conter somente as transições entre fácies sísmicas. Para tal, utilizamos somente duas classes: 0 e 1. A classe 0 (zero) compreende um local sem transição entre as fácies sísmicas e a classe 1 (um) representa um local de transição. Como as imagens de saída preditas pela rede DNFS deveriam conter somente cor branca e preta, isso poderia levar a uma diminuição não somente do número de filtros, mas também no número de camadas de conexão entre as partes *encoder* e *decoder*. Com isso mente, adotamos uma única camada de convolução 1D nessa conexão na DNFS. Entre essas partes, tanto na StNet como na U-Net existem duas camadas de convolução 1D que as interligam, na DNFS, uma camada. Assim, isso nos proporcionou menos parâmetros a serem ajustados e um reduzido tempo durante o treinamento dessa rede.

Entre as três arquiteturas utilizadas, foi observado que há uma transição mais acentuada entre os números de filtros dos blocos convolucionais e convolução transposta na StNet, foi de 128 para 1024 filtros, uma transição de oito vezes na quantidade de filtros, como pode ser visto na Figura 29, em DNFS, de 8Ω para 16Ω , duas vezes maior e em U-Net, de 512 para 1024, duas vezes maior como em DNFS. Isso leva a expansão do mapa de características do último bloco convolucional em 8 vezes ao se ligar com o bloco de transição, e a sua redução também em 8 vezes ao se ligar o bloco de transição com o primeiro bloco de convolução transposta. Devido a essa transição mais acentuada, observamos em nossos estudos de caso que as predições da StNet continham mais ruídos que as da DNFS e U-Net, ou seja, as imagens compostas pelas predições da StNet continham mais *pixels* pretos aleatórios que as advindas de DNFS e U-Net. Por isso, optamos por manter a taxa de transição entre número de filtros dos blocos de convolução e convolução transposta em duas vezes o valor do bloco anterior, como por ser visto na Figura 28.

4.2 Função de perda

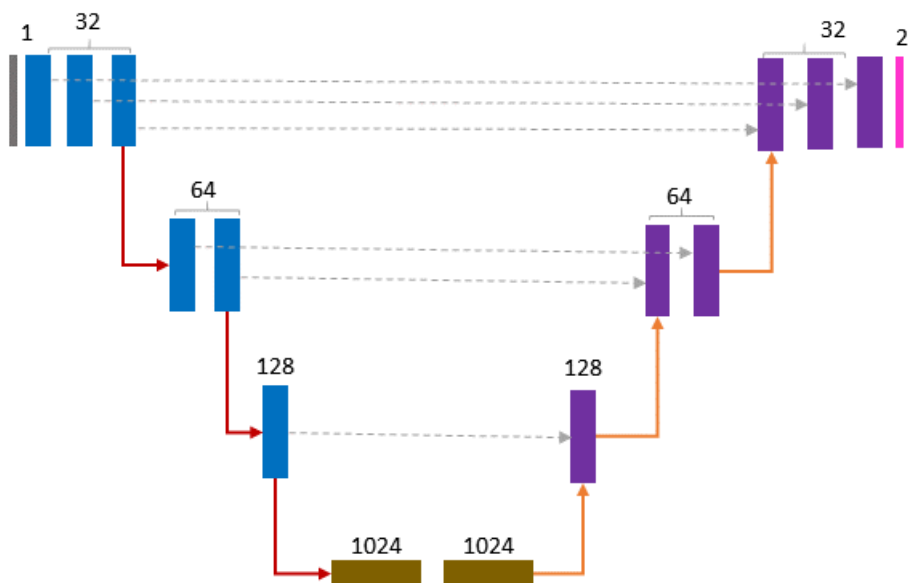
Em nosso trabalho, usamos uma função de perda composta. A qual é uma combinação linear das funções *cross-entropy* (Eq. (4.1)) e *Jaccard loss* (Eq. (4.2)). A *cross-entropy* foi responsável por atribuir uma de duas classes para cada pixel: 0 (zero) ou 1 (um). A classe zero (0) representa o fundo branco e a classe (1) um representa a transição entre as geométricas das fácies sísmicas. A responsabilidade da *Jaccard loss* foi medir a interseção entre as imagens previstas da rede

Figura 28: Esquema da variação de parâmetros da rede DNFS



Fonte: Criada pelo autor.

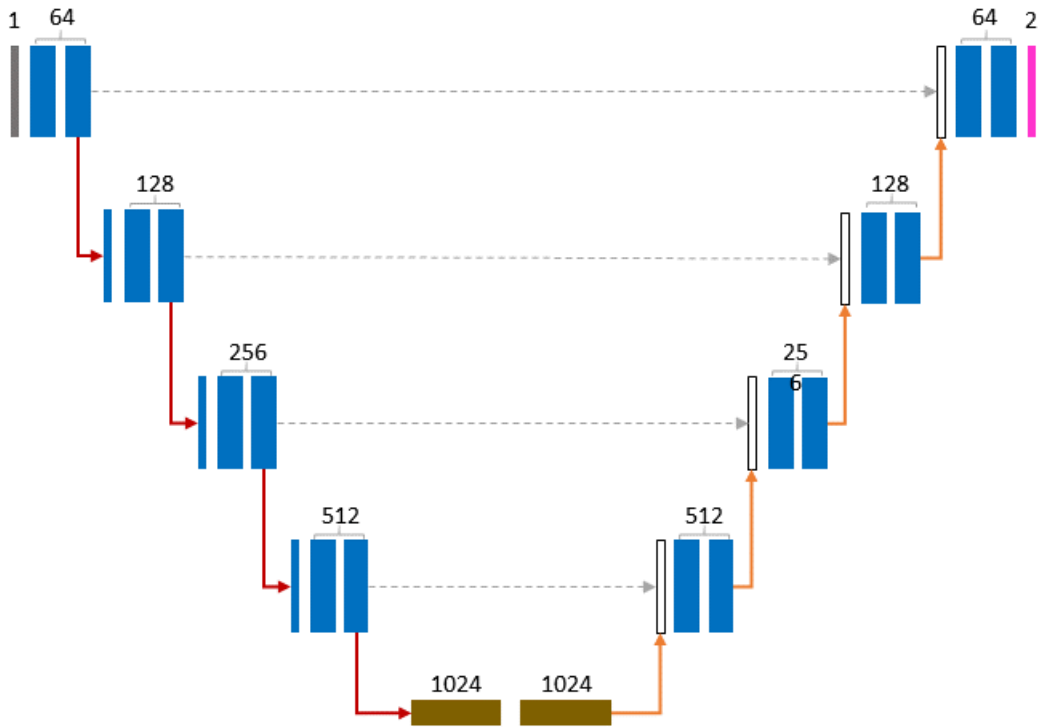
Figura 29: Estrutura da rede StNet.



Fonte: Adaptado de Di, Gao e AlRegib (2019b).

neural e suas máscaras correspondentes. Cada máscara em nosso trabalho representa partes de imagens sísmicas interpretadas, como mostramos na Seção 4.3. A *cross-entropy* pode ser

Figura 30: Estrutura da rede U-Net.



Fonte: Adaptado de Ronneberger, Fischer e Brox (2015).

definida como:

$$CrossEntropy = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C y_i^c \log s_i^c, \quad (4.1)$$

onde N é o número de *pixels* na imagem, C o número de classes, y_i^c é o indicador binário (0 ou 1) se a classe c é a classificação correta para o pixel i , e s_i^c é a probabilidade prevista pela rede. A *Jaccard loss*, por outro lado, pode ser definida como:

$$Jaccard = 1 - \frac{I(X)}{U(X)}, \quad (4.2)$$

onde $I(X)$ e $U(X)$ são definidos pelas Equações (4.3) e (4.4), respectivamente. Nessas equações, V representa o conjunto de todos os *pixels* de todas as imagens no conjunto de treinamento, X representa a saída da rede (fora da camada sigmóide) denotando probabilidades de pixel sobre o conjunto V , e $Y \in [0, 1]$ é a atribuição corretas possíveis para o conjunto V , onde 0 representa o pixel de fundo e 1 representa o pixel de linha preta.

$$I(X) = \sum_{v \in V} X_v * Y_v, \quad (4.3)$$

$$U(X) = \sum_{v \in V} (X_v + Y_v - X_v * Y_v), \quad (4.4)$$

Combinamos a *cross-entropy* (Eq. (4.1)) e a *Jaccard loss* (Eq. (4.2)) porque sozinhas não são suficientes para nosso propósito. A *cross-entropy* é útil para atribuir classes às posições dos *pixels*, mas não considera a distribuição espacial da classe. A *Jaccard loss*, por outro lado, torna-se útil quando o objetivo é medir a semelhança entre duas imagens. É necessário atribuir classes corretas a cada pixel e considerar a distribuição espacial dessas classes atribuídas para nosso problema. Assim, a seguir propomos uma função de perda composta que combina linearmente a *cross-entropy* e *Jaccard loss*, conforme mostrado a seguir:

$$Loss = \psi * CrossEntropy + (1 - \psi) * Jaccard, \quad (4.5)$$

onde ψ pesa a importância da *cross-entropy* e da *Jaccard loss*. Nossa função de perda calcula o erro entre as previsões e as máscaras contendo as interpretações, o que resulta no valor da perda se retropropagado através da rede para ajustar os pesos dos *kernels*. Como resultado, nossa função penaliza apropriadamente as previsões nas quais as classes de *pixels* inferidas por meio de *cross-entropy* não são semelhantes à distribuição espacial esperada das classes nas máscaras.

4.3 Dados de treinamento

Os dados utilizados para o treinamento das redes DNFS, StNet e U-Net, vieram da adaptação de dois datasets: *StData-12* e *Facies-Masks*. O *StData-12* foi apresentado por (DI; GAO; ALREGIB, 2019b) para o treinamento da StNet e é composto por seções sísmicas de um volume sísmico. Já o *Facies-Masks* foi apresentado por ALFARRAJ (2019) e possui a proposta de servir como base para a verificação do desempenho de redes voltadas a segmentação de múltiplas fácies sísmicas. Esses dois *datasets* são baseados no volume *Offshore F3 Block* (TERRANUBIS, 2020), mas possuem máscaras com diferentes interpretações dos dados sísmicos. As Figuras 31 e 34 apresentam exemplos de interpretações contidas em *StData-12* e em *Facies-Masks*. É importante se frisar que *datasets* com dados sísmicos anotados para segmentação são escassos pois são de grande importância para a indústria de petróleo e dependem de especialistas em interpretação sísmica para a sua criação. O *dataset StData-12* foi escolhido por fazer parte do trabalho de DI; GAO; ALREGIB (2019b), o qual apresenta a rede StNet que é utilizada como rede para comparação dos resultados obtidos neste trabalho. Para o *dataset Facies-Mask*, o motivo foi dele ser utilizado para a aplicação de diferentes redes neurais com a proposta de identificação de fácies sísmicas e de se propor como *benchmark* para os valores de índice IoU atingidos por redes na segmentação de fácies sísmicas.

As imagens contidas nos volumes do *StData-12* e *Facies-Masks* foram exportadas e executamos um algoritmo para a detecção das posições de mudança entre as cores utilizadas nas imagens, as mudanças foram preenchidas com bordas de *pixels* pretos e as áreas de não transição recebiam *pixels* brancos. Esses *pixels* pretos passaram a representar as interfaces entre as fácies sísmicas. O *dataset StData-12* é armazenado em arquivos do tipo SEG Y (W; FAICHNEY, 2002) e o *Facies-Masks* em arquivos exportados pela biblioteca numpy.

O Algoritmo 1 apresenta os passos para exportação das imagens com fundo branco e *pixels* pretos nas regiões onde há mudança de cor. Este algoritmo é utilizado pelos Algoritmos 2 e 3 que realizam a exportação das imagens dos *datasets*. Essa adaptação foi elaborada por dois motivos. Primeiro, as regiões que compõem as fácies sísmicas não necessariamente são compostas por texturas diferentes, assim, atribuir cores distintas a texturas não adjacentes, mas que poderiam pertencer ao mesmo tipo de fácies sísmicas impactaria a aprendizagem da rede. Segundo, como só há as cores branco e preto (duas classes) pode-se trabalhar com a segmentação binária e diminuir o tempo de treinamento. No Algoritmo 1 foi utilizado somente a técnica de *data agumentation*, a qual foi empregada em trabalhos relacionados como: SHI; WU; FOMEL (2018), MA et al. (2019), ZHANG et al. (2019), e em sistemas de segmentação de imagens médicas (RONNEBERGER; FISCHER; BROX, 2015), (CIRESAN et al., 2012). Utilizamos somente o *horizontal flipping* como técnica de *data agumentation* para que o dataset adaptado não alterasse as características do dado sísmico, já que as estruturas que formam as fácies sísmicas são dependentes entre si. Se usássemos *vertical flipping* ou rotação, uma linha divisória como na Figura 32 linha 5 e coluna 12 deixaria de compor o padrão que a deu origem. Já que nos dados sísmicos *vertical flipping* trocar a posição das amplitudes sísmicas sendo representadas e gera informações que podem não estar presentes na análise do intérprete do dado sísmico.

Algorithm 1 Algoritmo para exportação das imagens de treinamento

procedure WRITETRAININGIMAGES(*Imgs*, *Size*)

▷ *Imgs* Imagens a serem processadas.

▷ *Size*: dimensões dos segmentos a serem exportados.

for each *Img* ∈ *Imgs* **do**

P ← posições na imagem *Img* onde há mudança de cores

Imge ← imagem em cor branca ▷ Com dimensões iguais as de *Img*

Definir os *pixels* nas posições *P* de *Imge* com a cor preta

Segs ← segmentos de tamanho *Size* ∈ *Imge*

Gravar *Segs* ▷ Grava imagens com nomes aleatórios

Segs_{aug} ← *flipping* horizontal em *Segs*

Gravar *Segs_{aug}* ▷ Grava imagens com nomes aleatórios

Algorithm 2 Algoritmo de adaptação das imagens do dataset StData-12

V ← volume sísmico do dataset StData-12

D_{stdata} ← diretório para as imagens de treinamento

for each *secao_i* ∈ *V* **do**

D ← imagem com dados da seção *section_i* extraídos através da biblioteca *segvio*

Gravar dados da seção *section_i* em *D_{stdata}*

Gravar imagem da seção *section_i* em *D_{fm}*

D_{training} ← diretório para as imagens de treinamento

Imgs ← listar imagens gravadas em *D_{stdata}*

Size ← definição da largura e altura do segmento.

WRITETRAININGIMAGES(*Imgs*, *Size*)

Algorithm 3 Adaptação do dataset Facies-Masks

```

 $V \leftarrow$  volume do dataset Facies-Masks
 $D_{fm} \leftarrow$  diretório para as imagens de treinamento
for each  $section_i \in V$  do
     $D \leftarrow$  image com dados da seção  $section_i$  extraídos do volume
    Gravar imagem da seção  $section_i$  em  $D_{fm}$ 
 $D_{training} \leftarrow$  diretório para as imagens de treinamento
 $Imgs \leftarrow$  listar imagens gravadas em  $D_{fm}$ 
 $Size \leftarrow$  definição da largura e altura do segmento.
WRITETRAININGIMAGES( $Imgs, Size$ )
  
```

A Figura 31 apresenta um exemplo de imagem de saída fornecida pelo StData-12. Além desta, a figura também mostra a imagem de saída ajustada para este trabalho.

Figura 31: Imagens do *dataset StData* e a adaptação das máscara feita neste trabalho. À direita se encontra a imagem da seção sísmica de entrada, no meio a imagem rotulada de uma seção sísmica utilizada pelo trabalho de Di, Gao e AlRegib (2019a). A imagem mais à direita é a adaptada para o uso neste trabalho, a qual passou a ser constituída da cor branca para o fundo e de linhas separatórias entre as áreas que antes eram coloridas.

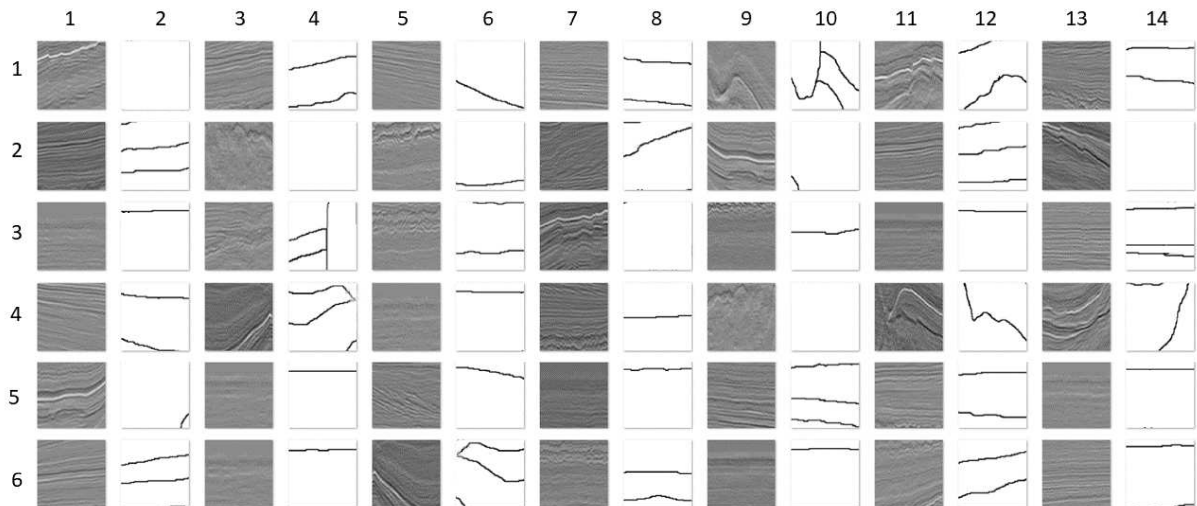


Fonte: Imagem central adaptada de Di, Gao e AlRegib (2019a).

A Figura 33, apresenta as imagens das seções sísmicas utilizadas para a criação do conjunto de dados para treinamento deste trabalho. As seções correspondem aos *inlines* 190, 290, 390 e 490 do volume sísmico *F3 Block Netherlands* que serviu de base para o *StData-12*. A Figura 34 exhibe algumas imagens do *dataset StData-12* utilizado para o treinamento das redes neurais. Observe que ambas as imagens possuem um único canal de cor, ou seja, são baseadas na escala de cinza onde o seu intervalo varia de 0 a 255. As imagens em escala cinza correspondem aos segmentos cortados das seções sísmicas exportadas, ao lado dela se encontram a máscara correspondente da fácies sísmica a ser segmentada pela rede neural. Uma informação importante sobre esse exemplo é que a região a ser segmentada é relacionada a algum padrão de fácies sísmicas, ou seja, a mudança entre as intensidades de cores vistas por uma pessoa não treinada para realizar interpretações sísmicas, não necessariamente corresponde ao que a rede neural deve aprender. Observe nas imagens que essa segmentação é complexa pois mesmo em meio a tantas mudanças de intensidades dos *pixels*, somente uma pequena parte é relevante e se refere a

alguma propriedade geológica de interesse do intérprete. O processo de criação destes segmentos é descrito no início desta seção.

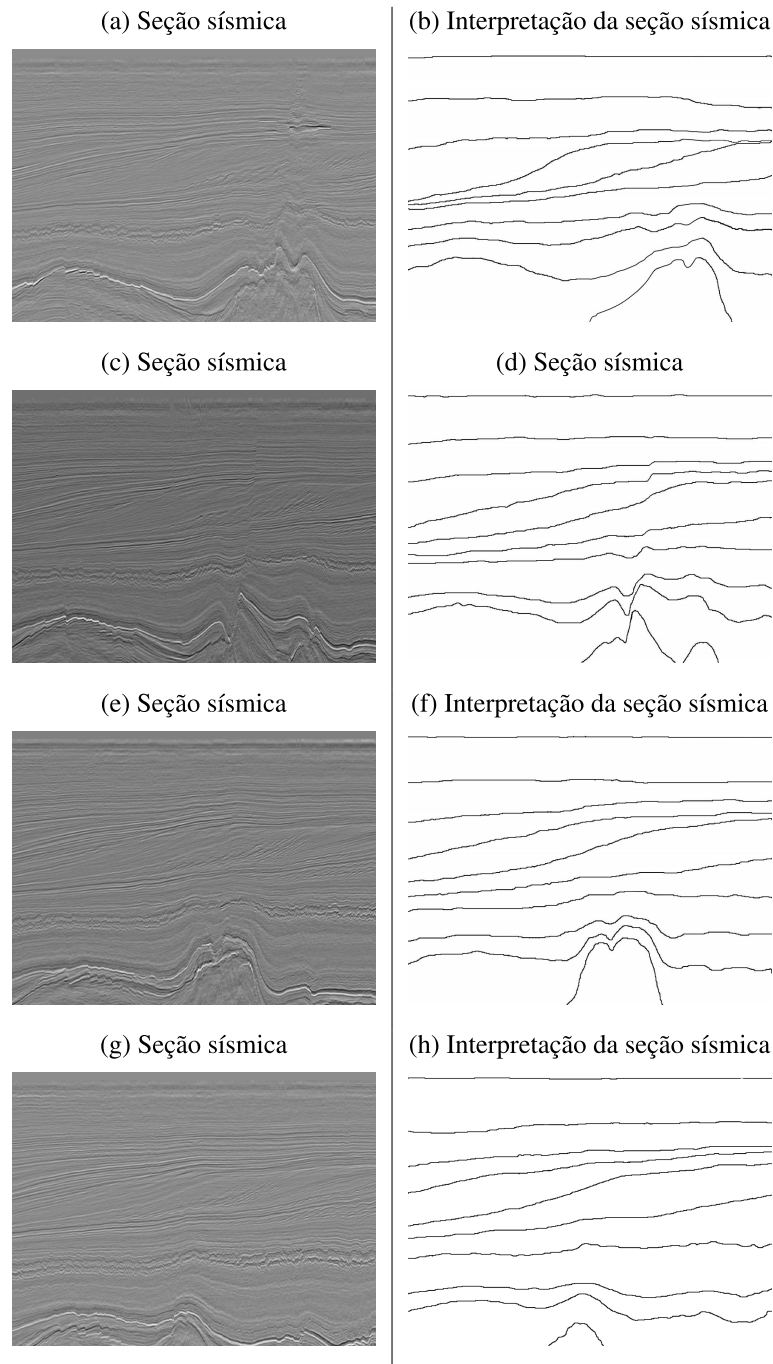
Figura 32: Exemplos de segmentos de imagem sísmica e máscara do *dataset* StData-12. As colunas ímpares correspondem aos segmentos adaptados a partir das seções sísmicas; as colunas pares correspondem aos segmentos das máscaras criadas para cada imagem adjacente à esquerda.



Fonte: Elaborada pelo autor a partir do *dataset* St-Data12 de Di, Gao e AlRegib (2019a).

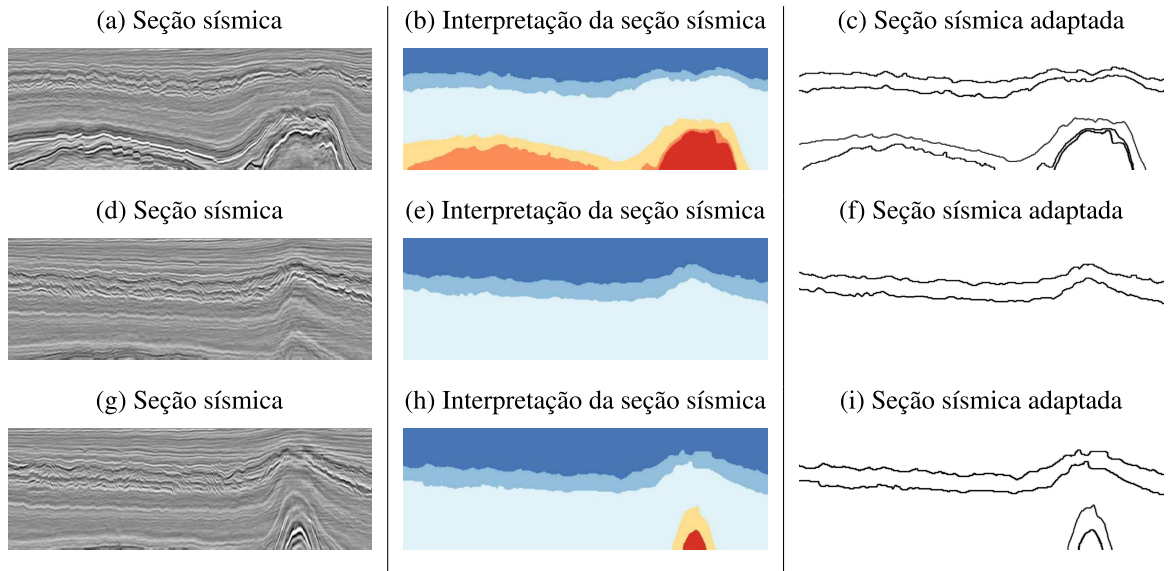
A Figura 34 exibe algumas seções sísmicas do *dataset* Facies-Masks adaptadas como foi apresentado na Seção 4.1. Observe que as interpretações deste *dataset* são diferentes das encontradas em *StData-12*. A Figura 35 exibe algumas imagens utilizadas para o treinamento das redes neurais. Nesta figura ambas as imagens possuem um único canal de cor, ou seja, são baseadas na escala de cinza onde o seu intervalo varia de 0 a 255. As imagens em escala cinza correspondem aos segmentos cortados das seções sísmicas exportadas, ao lado dela se encontram a máscara correspondente da fácies sísmica a ser segmentada pela rede neural.

Figura 33: Seções sísmicas do *dataset* StData-12. À direita se encontra a imagem da seção sísmica de presente no *dataset* StData-12. A imagem mais à direita é a adaptada para o uso neste trabalho, a qual passou a ser constituída da cor branca para o fundo e de linhas separatórias entre as áreas que antes eram coloridas.



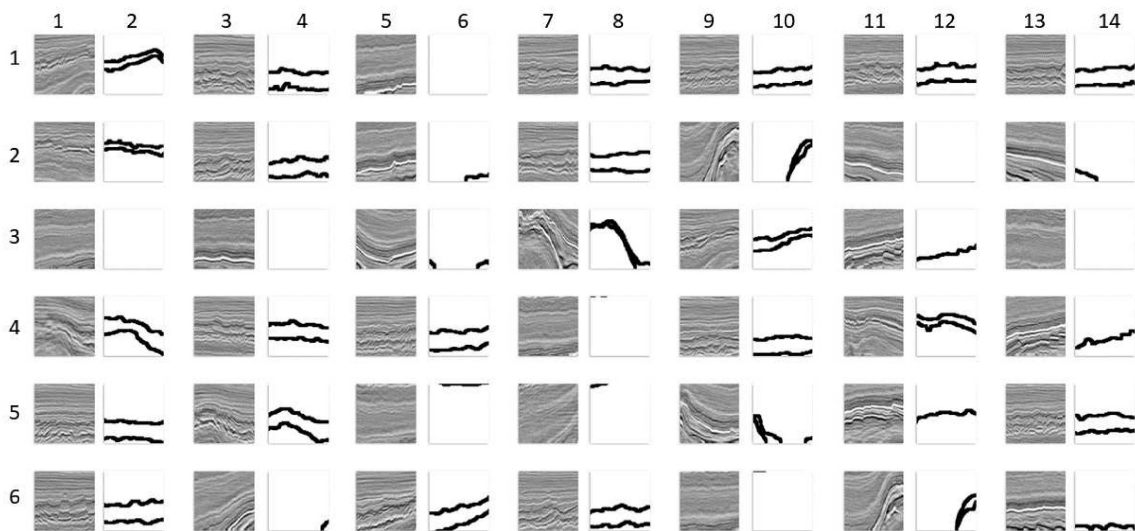
Fonte: Elaborada pelo autor a partir do *dataset* St-Data12 de Di, Gao e AlRegib (2019a).

Figura 34: À direita se encontra a imagem da seção sísmica de entrada, no meio a imagem rotulada de uma seção sísmica do *dataset* Facies-Mark. A imagem mais à direita é a adaptada para o uso neste trabalho, a qual passou a ser constituída da cor branca para o fundo e de linhas separatórias entre as áreas que antes eram coloridas.



Fonte: Elaborada pelo autor a partir do *dataset* Facies-Mark de Alfarraj (2019).

Figura 35: Exemplos de segmentos de imagem sísmica e máscara do *dataset* Facies-Mark. As colunas ímpares correspondem aos segmentos adaptados a partir das seções sísmicas; as colunas pares correspondem aos segmentos das máscaras criadas para cada imagem adjacente à esquerda.



Fonte: Elaborada pelo autor a partir do *dataset* Facies-Mark de Alfarraj (2019).

4.4 Treinamento

Foram treinadas as redes DNFS, StNet e U-Net com *batches* de imagens. Onde foi realizada a variação da quantidade de filtros para cada camada da rede DNFS. Cada *batch* pode ser entendido como um conjunto de imagens que são utilizadas para o treinamento da rede onde o valor da acurácia e da perda são calculados somente depois que essas imagens forem processadas pela rede neural; isso tem o efeito de acelerar o treinamento (DEAN et al., 2012).

Devido ao corte das imagens iniciais, ao final, foi feita a predição desses cortes e a junção dos resultados para se obter uma imagem inteira para fins de comparação. Durante o experimento foi utilizada uma função de perda criada a partir da combinação da função *Cross-entropy* e *Jaccard*, cada uma contribuindo com uma determinada porcentagem do cálculo do valor de perda. O treinamento das redes foi feito baseando-se no Algoritmo (4).

Algorithm 4 Treinamento rede neural

```

 $T_r \leftarrow$  dados de treinamento
 $V \leftarrow$  dados de validação
 $T_e \leftarrow$  dados de teste
 $Menor_{loss} \leftarrow$  dados de teste
 $Models \leftarrow$  modelos da rede DNFS neurais com os multiplicadores [4, 8, 16, 32, 64], StNet,
U-Net
for each  $m \in Models$  do
  for  $epoch \leftarrow 0, epochs$  do
    descongelar os pesos do modelo  $m$  para treinamento
    for each  $batch \in \mathcal{T}_r$  do
      treinar a rede
      extrair os resultados da predição
       $T_{loss} \leftarrow$  calcular a perda entre a imagem predita e a real
      atualizar o modelo  $m$ 
    congelar os pesos do modelo  $m$  para não haver treinamento
     $Loss_{acumulado} \leftarrow \inf$ 
    for each  $batch \in \mathcal{V}$  do
      prever a imagem de saída
       $V_{loss} \leftarrow$  calcular perda entre a imagem predita e a real
       $Loss_{acumulado} \leftarrow Loss_{acumulado} + V_{loss}$ 
     $Loss_{acumulado} \leftarrow \frac{Loss_{acumulado}}{len(V)}$   $\triangleright len(V)$  é a quantidade de itens em  $V$ 
    if  $Menor_{loss}$  is inf or  $Menor_{loss} > Loss_{acumulado}$  then
      salva o modelo atual como o melhor
       $Menor_{loss} \leftarrow Loss_{acumulado}$ 
   $S \leftarrow$  imagem vazia
  for each  $batch \in \mathcal{T}_e$  do
     $I_{out} \leftarrow$  prever a imagem de saída
     $S \leftarrow S + I_{out}$   $\triangleright$  Cola a imagem predita em  $S$ 
   $Index_{black} \leftarrow$  calcular o índice de pixels pretos preditos corretamente
  exibir  $S$  e  $Index_{black}$ 

```

A variação na quantidade de parâmetros da rede DNFS teve como objetivo verificar a escalabilidade do número de parâmetros a serem utilizadas para que se obtivesse predições das linhas que representam as interfaces entre as fácies sísmicas. Sendo isto feito sem se realizar modificações na quantidade e nas conexões das camadas da DNFS.

4.5 Considerações

Este capítulo apresentou a arquitetura das redes DNFS, StNet e DNFS. Para a rede DNFS é descrito na Seção 4.1 o modo de variação dos seus parâmetros com objetivo de se encontrar uma configuração mínima de rede que consiga fornecer predições das interfaces entre as fácies sísmicas através de linhas pretas. Para isso foi criada a função de perda composta a partir da combinação linear entre *cross-entropy* e *Jaccard loss* (Seção 4.2), onde a cada uma foi dado um peso diferente a partir do coeficiente ψ . A adaptação dos *datasets StData* e *Facies-Mark* para a utilização de segmentação binárias das fácies sísmicas é feita através dos Algoritmos (2) e (3) apresentados na Seção 4.3. Esses dados foram utilizados para o treinamento das redes de acordo com o Algoritmo (4) apresentado na Seção (4.4). Para verificar a adequação do método proposto ao problema de segmentação binária de fácies sísmicas, realizamos uma avaliação experimental tanto da arquitetura da DNFS como dos coeficientes da função de perda composta no Capítulo 5.

5 AVALIAÇÃO EXPERIMENTAL E RESULTADOS

Esta seção descreve a metodologia empregada para executar nossos experimentos, incluindo como nosso conjunto de dados foi preparado para treinar as redes neurais, quais métricas foram empregadas e como definimos o melhor coeficiente para a função de perda composta (Eq. (4.5)). Esta seção também mostra imagens previstas, tempo de treinamento, tamanho do disco e o número total de StNet e U-Net, assim como a quantidade de parâmetros das variantes de DNFS. Por fim, apresentamos e discutimos os principais resultados.

5.1 Metodologia

Dividimos os *datasets* StData (DI; GAO; ALREGIB, 2019b) e Facies-Mark (ALFARRAJ, 2019) em conjuntos de dados de treinamento, validação e teste. Para o *dataset* StData foram 768, 192 e 96 imagens, e para o Facies-Mark, 2136, 270, 62 respectivamente. Nossos testes usaram as redes DNFS, StNet e U-Net. A estrutura dessas redes foi mantida, mas variamos o número de filtros em cada camada da DNFS, conforme discutido na Seção 4.1. Usamos as imagens sísmicas como entradas para as redes e obtivemos previsões de saída onde cada posição de pixel (i, j) na saída era associada a uma classe que corresponde à cor branca ou preta. A cor preta foi associada com a interface de separação entre fácies sísmicas e a cor branca com as regiões contidas entre essas interfaces. Cada rede neural recebeu as mesmas informações do conjunto de dados como entrada e tinha parâmetros semelhantes para treinamento.

Usamos parâmetros de treinamento semelhantes para DNFS, StNet e U-Net para comparar as previsões dessas redes umas com as outras. Em seu trabalho, DI; GAO; ALREGIB (2019a) treinou a StNet com o otimizador Adam (KINGMA; BA, 2014) para minimização de perdas e RONNEBERGER; FISCHER; BROX (2015) treinou a U-Net com o *Stochastic Gradient* (BOTTOU, 2012). Neste trabalho, usamos o mesmo otimizador de StNet, pois este oferece convergência mais rápida em comparação com o *Stochastic Gradient* (DOGO et al., 2018). Treinamos as configurações da DNFS da Figura 28 com uma taxa de aprendizagem de 0,001 e um máximo de 500 *epochs*, utilizamos a técnica de *Early Stopping*, onde o treinamento da rede é interrompido quando uma métrica de avaliação fornece os mesmos valores por um determinado número de *epochs*. Para o *Early Stopping* selecionamos a métrica de IoU (Eq. (5.1)) e um número de 15 *epochs* para se interromper o treinamento, essa quantidade de *epochs* foi determinada a partir da análise dos gráficos com as curvas de aprendizagem das redes. Na etapa final de cada *epoch*, usamos o grupo de dados de validação para verificar se as redes estavam sobre-ajustadas. A rede U-Net usou um *kernel* 3x3 com um *stride* de 1 para as camadas presentes nas partes *encoder* e *decoder*, os quais também foram utilizados para a construção da DNFS e StNet. Não encontramos os valores reais de *stride* e *kernel* para a StNet porque eles não foram relatados em DI; GAO; ALREGIB (2019a).

De modo a avaliar o desempenho dos modelos, utilizamos como métricas a porcentagem

Tabela 15: Parâmetros de treinamento e tamanhos de rede neural para DNFS, StNet e U-Net. O asterisco se refere aos resultados das arquiteturas StNet e U-Net propostas em RONNEBERGER; FISCHER; BROX (2015) e DI; GAO; ALREGIB (2019a).

Rede	Fator	Parâmetros	Disco	Batch Size	Dropout	LR	<i>Epochs</i>
DNFS	4	86,181	1,02 MB	16	10%	0,001	500
DNFS	8	341,001	2,96 MB	16	10%	0,001	500
DNFS	16	1,356,561	11,00 MB	16	10%	0,001	500
DNFS	32	5,411,361	41,63 MB	16	10%	0,001	500
DNFS	64	21,615,681	165,24 MB	8	10%	0,001	500
StNet*	–	1,505,697	12,00 MB	8	10%	0,001	500
U-Net*	–	31,059,085	273,31 MB	8	10%	0,001	500

Fonte: Elaborada pelo autor.

de *pixels* pretos preditos corretamente, o tempo de treinamento e o uso de memória para cada configuração de rede para medir os seus desempenhos. A Tabela 15 apresenta os valores dos parâmetros e hiper-parâmetros. Os valores de *Batch size* foram determinados experimentalmente, de modo que todas as redes pudessem utilizar o mesmo conjunto de dados de treinamento e serem treinadas na mesma GPU. O valor de 10% para o *dropout* foi escolhido ao variarmos a porcentagem de 0 a 90% e verificarmos que com 10%, as redes forneceram os maiores valores de índice IoU (veja Tabelas 17, 18, 19 no Apêndice). O valor de *learning rate* (LR) de 0,001 foi escolhido ao verificarmos que valores maiores faziam a rede não conseguir aprender a prever as linhas pretas (veja Tabela 20 no Apêndice). Para o valor de 500 *epochs* observamos que se a rede chegasse a esse número, haveria um sobre ajustamento aos dados de treinamento. Durante a execução de cada *epoch*, aplicamos uma métrica para avaliar as previsões da rede neural. Essa métrica é chamada de Intersecção sobre União (IoU - Intersection over Union) (ZHOU et al., 2019) e é definida como:

$$IoU_{metric} = \frac{|p \cap m|}{|p \cup m|}, \quad (5.1)$$

onde p é a previsão da rede e m a máscara correspondente no conjunto de dados. Esta equação mede o quanto duas imagens se sobrepõem, sendo útil para avaliar os resultados da segmentação sísmica. Verificamos, experimentalmente, que o mínimo adequado do valor fornecido por essa métrica é de 0.95. No entanto, como nosso problema diz respeito principalmente aos *pixels* pretos relacionados às interfaces entre as fácies sísmicas, adotamos uma métrica adicional. Essa métrica compara os resultados previstos de nossas redes modificadas, levando em consideração apenas os *pixels* pretos, conforme mostrado abaixo:

$$P_{bp}(T, P) = \frac{S_{bp}(T, P)}{\sum_{i=0}^{w-1} \sum_{j=0}^{h-1} T(i, j)}, \quad (5.2)$$

onde T representa a imagem da real da máscara da interpretação sísmica, P a imagem prevista, w e h a largura e altura das imagens, $B_p(T, P)$ representa uma recompensa por predição correta de *pixel* preto, $P_{bp}(T, P)$ a soma de todos os valores de $B_p(T, P)$ como mostrado na Equação (5.4),

e $P_{bp}(T, P)$ representa a porcentagem de previsões corretas de *pixels* pretos conforme mostrado na Equação (5.3). Verificamos, experimentalmente, que o mínimo adequado do valor fornecido por essa métrica é de 50% de acertos.

$$S_{bp}(T, P) = \sum_{i=0}^{w-1} sum_{j=0}^{h-1} B_p(T, P) \quad (5.3)$$

$$B_p(T, P) = \begin{cases} 1, & \text{if } T(i, j) = 1 \wedge P(i, j) = 1 \\ 0, & \text{caso contrário} \end{cases} \quad (5.4)$$

A lógica por trás da combinação das métricas de comparação IoU (Eq. (5.1)) e de *pixels* pretos (Eq. (5.2)), a arquitetura da rede DNFS e as máscaras em preto e branco (Seção 4.3) é baseada em dois aspectos. Primeiro, elas nos permitem diminuir a quantidade de dados a ser usada para treinar a rede. Segundo, permitem realizar a segmentação de múltiplas fácies sísmicas simultaneamente, ao contrário da maioria dos trabalhos relacionados que usam classes diferentes para cada fácies sísmica ou se concentram na segmentação de um único tipo de fácies sísmica. Para a função de perda composta (Eq. (4.5)) utilizada para o treinamento das redes, devido a função de perda *cross-entropy* não levar em consideração a distribuição espacial das classes ao calcular o valor de perda, fizemos a sua combinação linear com a *Jaccard loss*, sendo esta a responsável por verificar a distribuição espacial.

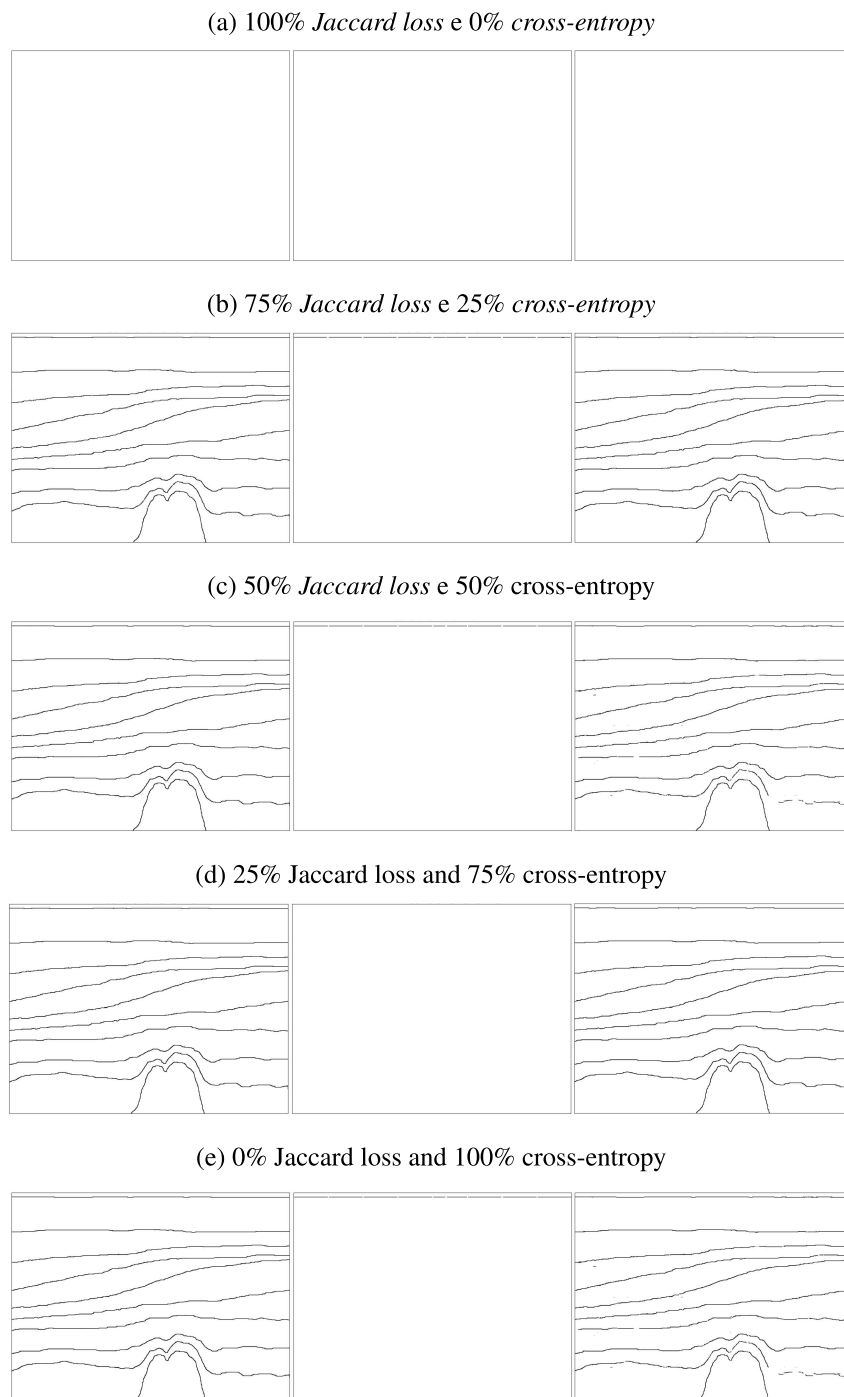
O treinamento da rede neural e os cálculos métricos foram feitos com uma infraestrutura fornecida por *Keras functional API* para *TensorFlow 2.0* (ABADI et al., 2015) e *Google Colab Pro* (LC, 2018). Para ler os arquivos *SEG-Y* (W; FAICHNEY, 2002) com dados sísmicos brutos, usamos a biblioteca *seggyio python* (ASA, 2019) e criamos imagens das seções sísmicas com *matplotlib* versão 3.2.2 (CASWELL et al., 2020).

5.2 Otimização do coeficiente de perda

Esta seção detalha como definimos o melhor valor para o coeficiente da função de perda utilizando como base o *dataset StData-12*. Vale lembrar que o coeficiente ψ da Equação (4.5) corresponde ao peso dado à *cross-entropy* em comparação à *Jaccard loss* em nossa função de perda composta. Para tanto, realizamos experimentos com $\psi = \{0,0, 0,25, 0,50, 0,75, 1,0\}$. Em todos os casos, definimos DNFS com fator 8, pois foi a variante de DNFS que com o menor número de parâmetros foi capaz de fornecer previsões das linhas pretas, e utilizamos a StNet e U-Net para medir a porcentagem de *pixels* pretos aplicando a Equação (5.2) nas imagens preditas. Escolhemos esta configuração porque o tempo de treinamento foi curto (cerca de 10 minutos) e a porcentagem correta de *pixels* pretos foi próximo de 60%, o que foi uma ocorrência comum em nossos testes. As imagens preditas estão presentes na Figura 36. Como visto, o coeficiente de 0,75 entregou os melhores resultados para nosso tipo de dados e arquitetura de rede. Portanto, no que segue, nossos resultados consideram $\psi = 0,75$.

Nas Figuras 37 e 38 podemos observar que o valor fornecido pela métrica IoU durante o treinamento das redes neurais DNFS, StNet e U-Net varia logo nos primeiros *epochs*, o que nos informa que a rede inicialmente se ajusta para a predição das cores brancas que representam as áreas entre as fácies sísmicas. Isso se deve ao fato de que a quantidade de *pixels* brancos em relação a de pretos é cerca de mais de 97% de diferença. Assim, primeiramente a rede se ajusta para a predição destes e somente quando a curva de aprendizagem se aproxima do achatamento é que ela começa a aprender a como realizar as predições das linhas pretas separatórias das fácies sísmicas. Mesmo com o alto desbalanceamento entre as classes 0 (*pixel* branco) e 1 (*pixel* preto) a utilização da função de perda *Jaccard loss* combinada com a *cross-entropy* foi capaz de treinar as redes. Porém, o uso da métrica IoU sozinha, pode levar a ideia de que a rede a rede conseguiu valores acima de 98% de acertos nas predições. Porém, pelo que sugere a Tabela 16, isso não representa que a rede será capaz de realizar predições das linhas pretas. Por isso, utilizamos a Equação (5.2) durante os testes para verificar o real valor de *pixels* pretos preditos corretamente. Na figura 37(a), a DNFS com fator 4 necessitou de cerca de 50 *epochs* para atingir aproximadamente o valor de 0,1 de *Jaccard distance* e em cerca de 225 *epochs* o seu treinamento foi encerrado por não haver diminuição do *Jaccard distance*. Como essa é a variação da DNFS com menos parâmetro, temos que para este *dataset* seja necessário uma variante da DNFS com um maior número de parâmetros e que seja treinada por mais *epochs*, sendo sendo isso apoiado pelos resultados apresentados na Tabela 16. A figura 37(b), para a DNFS com fator 8, diferentemente da DNFS com fator 4, obteve tanto a curva de aprendizado como de validação mais próximas entre si no início do treinamento. Em cerca de 50 *epochs*, como também pode ser verificado nas Figuras 37(c), 37(d), 38(a), 38(c) as curvas ficaram próximas entre si. Já a StNet na Figura 38(b) e a DNFS na Figura 37(a) apresentam curvas no qual o índice *Jaccard distance* ficou próximo a 0,05 e fez com que essas redes não fossem capazes de fornecer predições das linhas pretas, conforme podemos ver na Tabela 16.

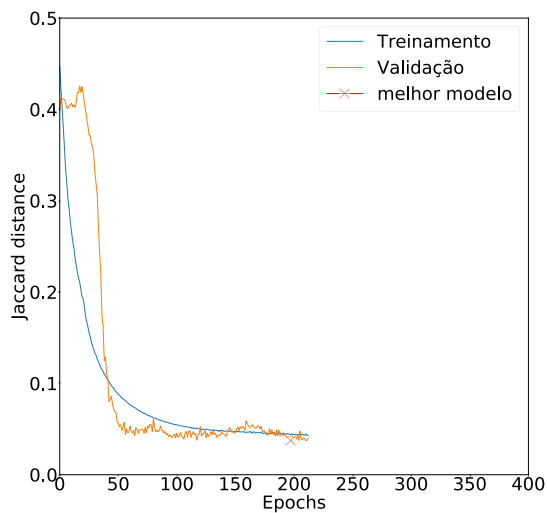
Figura 36: Comparação das previsões feitas pelas redes DNFS (com fator 8), StNet e U-Net com diferentes valores do coeficiente ψ para a função de perda composta (Eq.(4.5)).



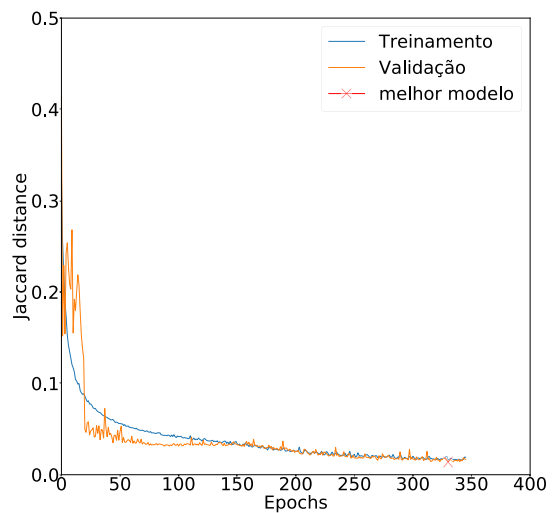
Fonte: Elaborada pelo autor a partir do *dataset* St-Data12 de Di, Gao e AlRegib (2019a).

Figura 37: Curvas de aprendizagem para DNFS com fator 4, 8, 16 e 32. Evolução do índice *Jaccard* (eixo vertical) durante o treinamento das redes DNFS, StNet e U-Net com a progressão dos *epochs* (eixo horizontal).

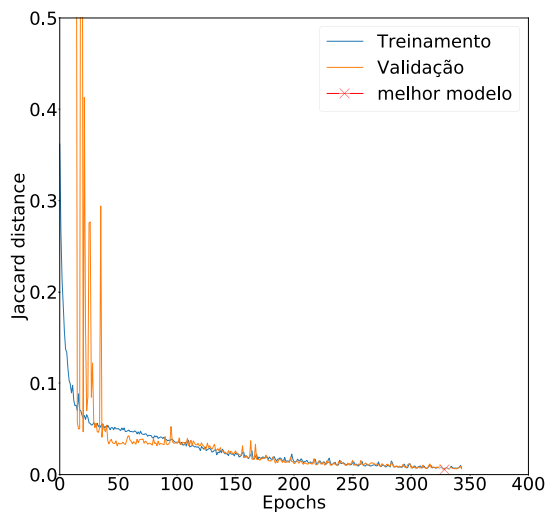
(a) DNFS com fator 4



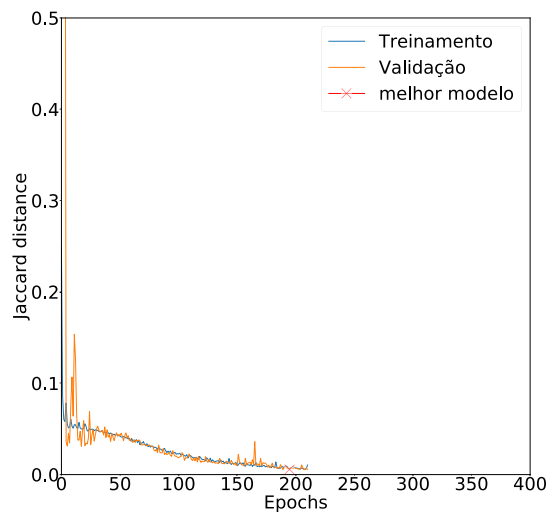
(b) DNFS com fator 8



(c) DNFS com fator 16

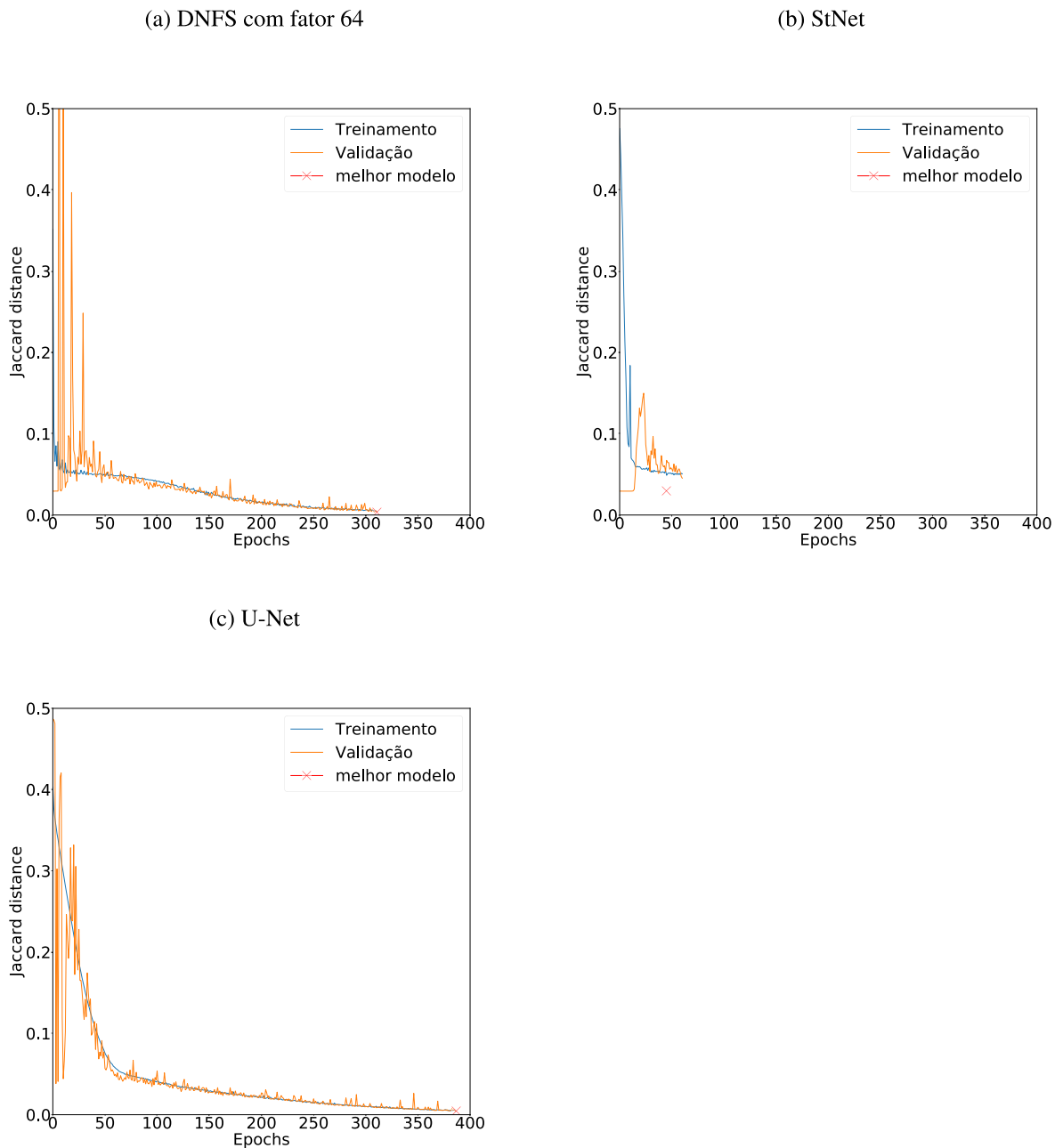


(d) DNFS com fator 32



Fonte: Criado pelo autor.

Figura 38: Curvas de aprendizagem para DNFS com fator 64, StNet e U-Net. Evolução do índice *Jaccard* (eixo vertical) durante o treinamento das redes DNFS, StNet e U-Net com a progressão dos *epochs* (eixo horizontal).



Fonte: Criado pelo autor.

5.3 Resultados e discussão

Nosso objetivo com a variação dos parâmetros da DNFS era encontrar uma configuração de rede *encoder-decoder* mínima quando comparada a U-Net e StNet capaz de oferecer predições das interfaces entre as fácies sísmicas em menor tempo que essas duas redes. Assim, essa rede poderia ser inserida em um fluxo diário de trabalho de um intérprete de dados sísmicos. Obtivemos essa configuração comparando os resultados previstos de cada rede usando a porcentagem calculada pela Equação (5.2). As Figuras 40, 41, 42, 43, 44, 45, 46 mostram as saídas preditas para o *dataset StData12*, as Figuras 47, 48, 49, 50, 51, 52 para o *dataset* para o *Facies-Mark* e a Tabela 16 apresenta os resultados experimentais para tempo de treinamento, índice IoU e porcentagem de *pixels* pretos preditos corretamente para as variantes da DNFS e as redes StNet e U-Net.

Tabela 16: Comparação de desempenho de DNFS, StNet e U-Net, em termos de tempo de treinamento, precisão e porcentagem de *pixels* pretos corretos. O melhor resultado da arquitetura DNFS (quando combinamos o tempo de treinamento e a porcentagem de *pixels* pretos) é destacado em negrito.

Rede	Fator	Treinamento (H:m:s)	Parâmetros	Disco	IoU(%)	<i>Pixels</i> pretos(%)	<i>Dataset</i>
DNFS	4	00:02:35	86,181	1,02 MB	99,27	0,0	<i>StData-12</i>
DNFS	8	00:05:16	341,001	2,96 MB	98,95	51,00	<i>StData-12</i>
DNFS	16	00:08:16	1,356,561	11,00 MB	99,08	58,80	<i>StData-12</i>
DNFS	32	00:09:58	5,411,361	41,63 MB	99,10	60,23	<i>StData-12</i>
DNFS	64	00:36:22	21.615.681	165,24 MB	99,13	61,54	<i>StData-12</i>
StNet	–	00:02:31	1,505,697	12,00 MB	99,27	0,0	<i>StData-12</i>
U-Net	–	00:49:29	31,059,085	273,31 MB	99,18	63,42	<i>StData-12</i>
DNFS	4	00:15:44	86,181	1,02 MB	95,90	61,94	<i>Facies-Mark</i>
DNFS	8	00:16:01	341,001	2,96 MB	95,56	61,12	<i>Facies-Mark</i>
DNFS	16	00:16:24	1,356,561	11,00 MB	95,78	57,88	<i>Facies-Mark</i>
	32	00:16:21	5,411.361	41,63 MB	96,16	63,66	<i>Facies-Mark</i>
DNFS	64	00:20:06	21,615,681	165,24 MB	95,62	59,17	<i>Facies-Mark</i>
StNet	–	01:00:46	1,505,697	12,00 MB	96,17	65,48	<i>Facies-Mark</i>
U-Net	–	00:47:45	31,059,085	273,31 MB	96,25	64,71	<i>Facies-Mark</i>

Fonte: Criada pelo autor.

Como pode ser verificado na Tabela 16, as redes DNFS e U-Net apresentam valores de índice de porcentagem de *pixels* pretos mais altos que a StNet para o *dataset StData*. Os dados na Tabela 16 revelam que DNFS com fator 32 (DNFS₃₂) teve o melhor resultado quando juntamos a porcentagem de *pixels* pretos e o tempo de treinamento. Esta rede levou cerca de 10 minutos para treinar. Quando consideramos o tamanho de disco para DNFS, StNet e U-Net, vimos uma grande diferença. A StNet e a U-Net ocupam mais espaço em disco do que a mínima variante de DNFS com o fator 8 (DNFS₈), a qual foi capaz de oferecer predições das interfaces entre fácies sísmicas, o que a StNet não foi. A DNFS₈ corresponde a 24% (1 MB) do tamanho em disco da StNet (12 MB) e apenas a 1,08% (1 MB) do tamanho em disco da U-Net (273 MB); e a rede

DNFS₃₂ ocupa 15,23% (41 MB) do tamanho em disco da U-Net (273 MB). Conseqüentemente, a DNFS₈ é muito menor que a StNet e U-Net, e mesmo como a DNFS₃₂, elas foram capazes de prover porcentagens de *pixels* pretos semelhantes a da U-Net. Outra característica da DNFS é que ela fornece uma pequena porcentagem (cerca de 10%) de aumento de *pixels* pretos mesmo quando aumentamos os valores dos fatores, o que conseqüentemente incrementa o número total de parâmetros. Isso sugere que nossa arquitetura de rede neural é mais adequada para segmentar fácies sísmicas em dados sísmicos usando o conjunto de dados adaptados apresentado na Seção 4.3 do que StNet e U-Net para o *dataset* StData-12.

Para o *dataset* *Facies-Mark* observamos que rede StNet passou a conseguir realizar a predição das interfaces entre as fácies sísmicas (as linhas pretas), mas para isso ela demorou 15 minutos a mais que a U-Net e 40 minutos a mais que a DNFS com fator 64 (DNFS₆₄), a que contém a maior quantidade de parâmetros das variantes da DNFS. Ou seja, a StNet foi a rede que levou mais tempo para ser treinada, foi 1,27 vezes mais lenta que a U-Net, 3,02 vezes mais que a DNFS₆₄, 3,7 vezes mais que a DNFS₃₂ e DNFS₁₆, DNFS₈ e 3,9 vezes mais que a DNFS₄. Porém, para este *dataset* foi a que atingiu o maior valor de porcentagem de *pixels* pretos preditos corretamente ao custo de um maior tempo de treinamento. Agora, se comparamos o valor de 65,48% atingido pela StNet com o valor de 63,66 % atingido pela DNFS₃₂, notamos que é uma diferença de apenas 1,82% ao custo de cerca de 44 minutos a mais de espera para o intérprete obter as imagens com as predições feitas através de redes neurais.

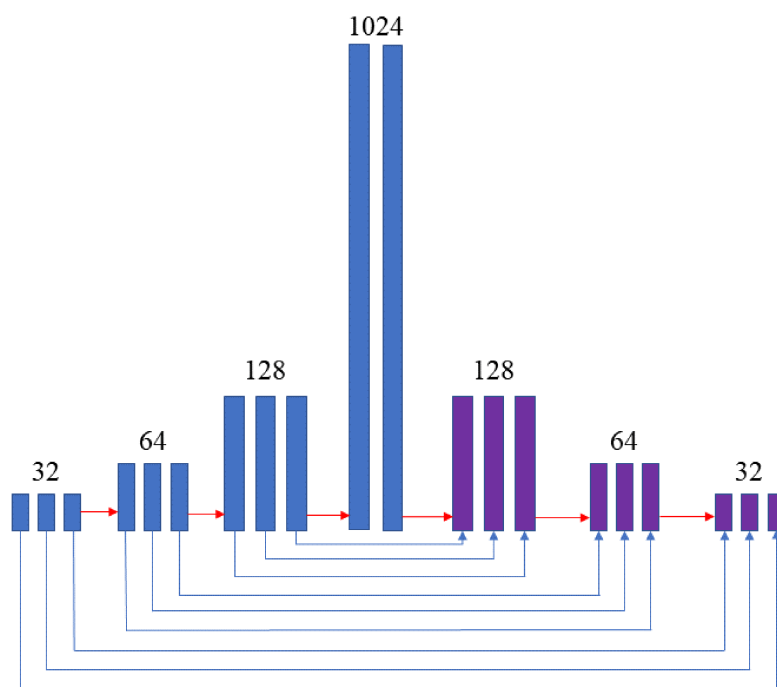
Embora a StNet e U-Net atingissem maiores valores de porcentagem de *pixels* pretos preditos corretamente que as da DNFS, nos quesitos tempo de treinamento e tamanho em disco, as variantes da DNFS são melhores. Estas variantes atingiram menores valores que as redes bases de comparação (StNet e U-Net) e forneceram predições das linhas separatórias das interfaces entre as fácies sísmicas. O que para a integração em um fluxo diário de trabalho, no qual o intérprete de dados sísmicos realiza a interpretação de seções sísmicas e as usa para o treinamento da rede, o tempo de espera para obtenção da segmentação das fácies sísmicas seria reduzido. Assim, a DNFS fornece ao intérprete a possibilidade de realizar diferentes interpretações sobre as seções sísmicas, e contar com uma rede neural que aplique as interpretações feitas em outras outras seções sísmicas.

Analisando o modo como a métrica da Equação (5.2) foi construída para a obtenção da porcentagem de *pixels* pretos preditos corretamente, percebemos que se um *pixel* for predito na posição $(i + 1, j + 1)$ ao invés da posição (i, j) em uma imagem 2D, onde i representa um índice da linha e j o da coluna. Esse *pixel*, com uma diferença de apenas uma posição, não seria considerado na porcentagem de *pixels* pretos preditos corretamente. Então, consideramos ser mais adequado medir a distância média dos *pixels* pretos preditos dos reais. Assim, uma *pixel* predito como perto de uma linha preta real seria melhor que um predito longe de qualquer linha. Com isso, a realização da adaptação da métrica da Equação (5.2) teria a objetivo de aumentar a porcentagem de *pixels* pretos preditos considerando um pequeno intervalo de deslocamento entre suas posições e esta ideia também poderia ser incorporada na função de perda composta (Eq.

(4.5)) que apresentamos na Seção 4.2. A investigação da aplicação dessa métrica, o melhor valor médio desse intervalo e a construção de uma nova função de perda são pontos de investigação a serem realizados futuramente.

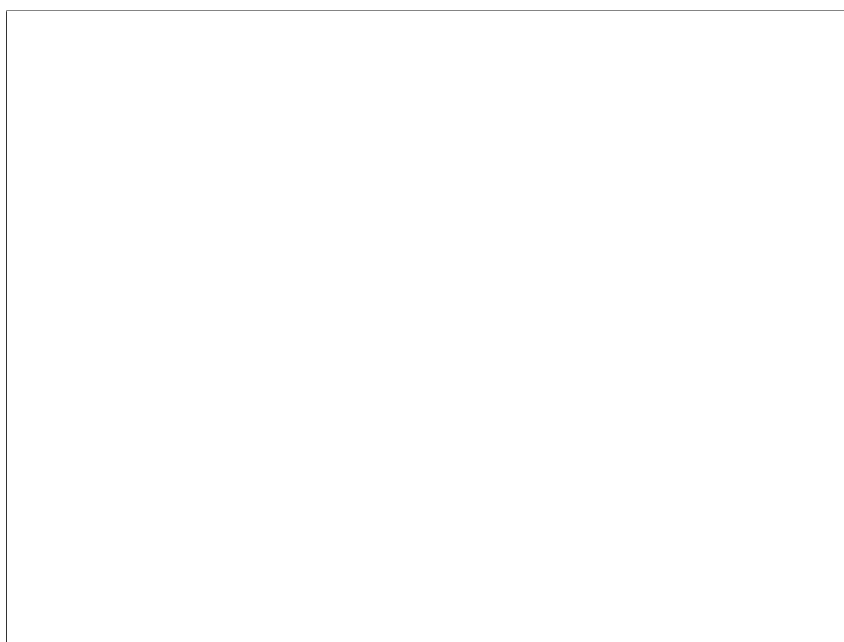
Entre as três redes testadas, foi observado que há uma transição mais acentuada entre os números de filtros dos blocos convolucionais e de convolução transposta na StNet, foi de 128 para 1024 filtros, uma transição de oito vezes na quantidade de filtros, como pode ser visto na Figura 39. Isso leva a expansão do mapa de características do último bloco convolucional em 8 vezes ao se ligar com o bloco de transição, e a sua redução também em 8 vezes ao se ligar o bloco de transição com o primeiro bloco de convolução transposta. Devido a essa transição mais acentuada, observamos que as predições da StNet continham mais ruídos que as da DNFS e U-Net, ou seja, as imagens compostas pelas predições da StNet continham mais *pixels* pretos aleatórios que as advindas de DNFS e U-Net. Além da mudança acentuada do número de filtros, o que não acontece com a DNFS (Figura 28) e U-Net (Figura 30), se nota que entre todas as camadas da rede StNet há transferências de propriedades (*skip-connections*) o que acreditamos ser a segunda razão que pôde levar a rede a criar ruídos nas camadas da parte do *encoder*, uma vez que as propriedades extraídas inicialmente foram passadas diretamente para outras camadas posteriores, sem qualquer filtro. Então, isso atrapalharia a convergência da rede e aumentaria o seu tempo de treinamento como aconteceu com a StNet.

Figura 39: Esquema de variação de mapas de propriedades na StNet e o uso de *skip-connections*. As barras azuis representam a quantidade de filtros nos blocos que compõem a StNet. As setas de cor azul as transferências de propriedades entre camadas internas de blocos e as vermelhas as transferências entre blocos.



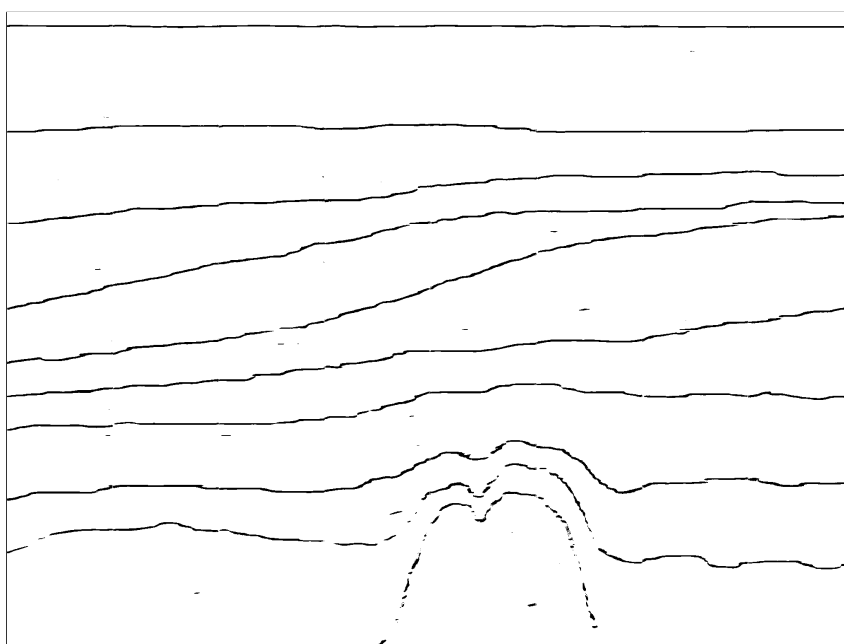
As Figuras 40, 41, 42, 43, 44, 45 e 46 exibem as previsões das variantes das redes DNFS e as previsões da StNet e U-Net para o *dataset StData-12*. Como pode ser observado nessas figuras, a DNFS foi capaz de fornecer melhores previsões das linhas pretas que representam as interfaces entre as fácies sísmicas, a partir do fator 8, o que sugere que para este *dataset* a configuração mínima da DNFS é baseada nesse fator.

Figura 40: Predição do *inline 390* do *dataset StData-12* com a rede DNFS com fator 4.



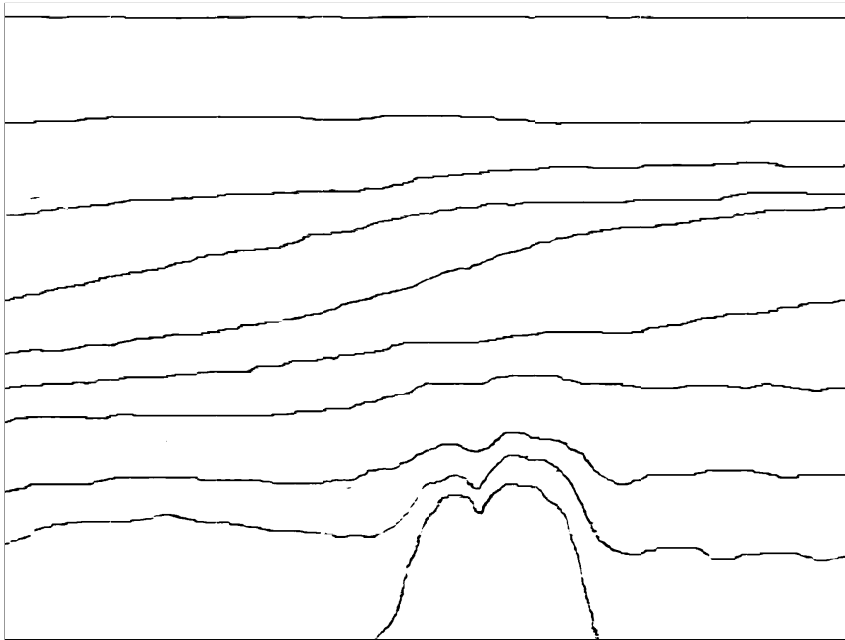
Fonte: Elaborada pelo autor a partir do *dataset St-Data12* de Di, Gao e AlRegib (2019a).

Figura 41: Predição do *inline 390* do *dataset StData-12* com a rede DNFS com fator 8.



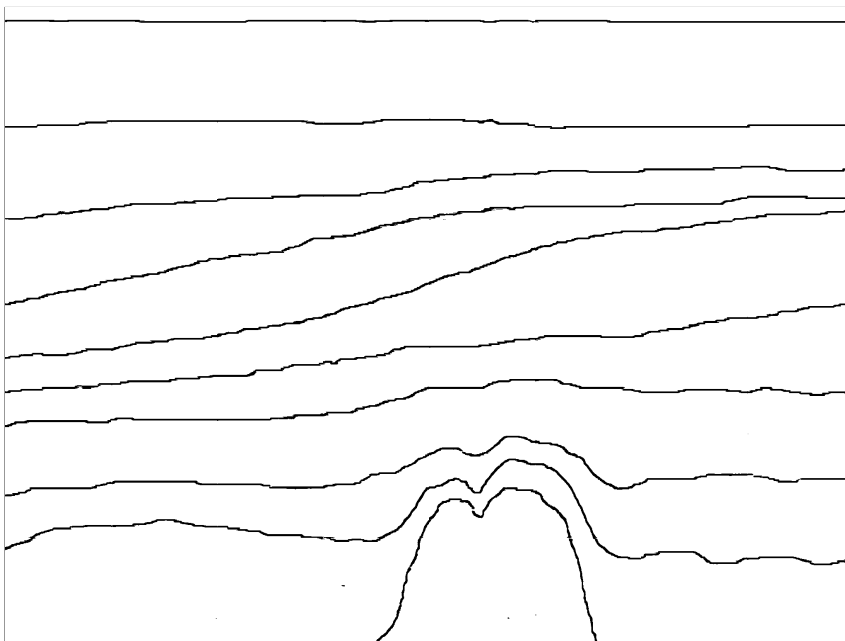
Fonte: Elaborada pelo autor a partir do *dataset St-Data12* de Di, Gao e AlRegib (2019a).

Figura 42: Predição do *inline* 390 do *dataset* StData-12 com a rede DNFS com fator 16.



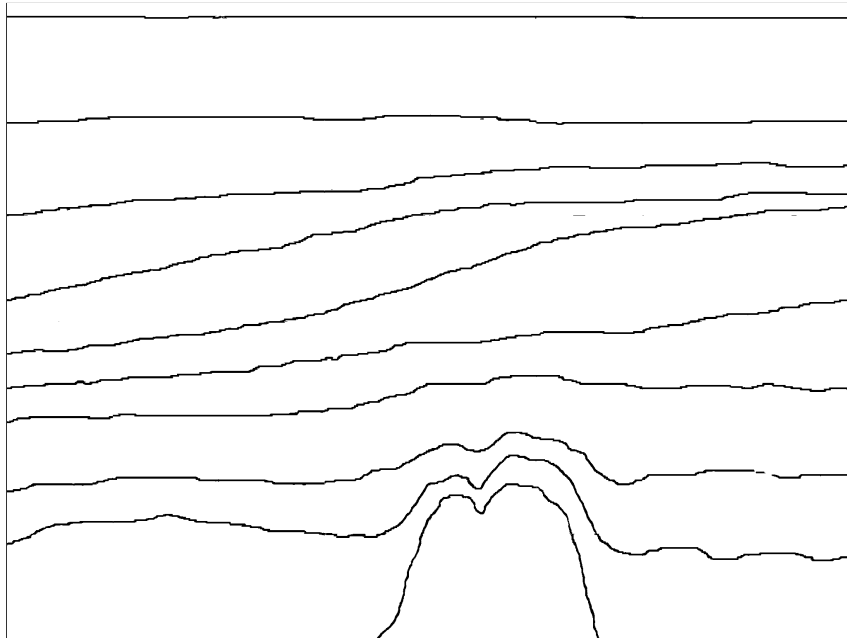
Fonte: Elaborada pelo autor a partir do *dataset* St-Data12 de Di, Gao e AlRegib (2019a).

Figura 43: Predição do *inline* 390 do *dataset* StData-12 com a rede DNFS com fator 32.



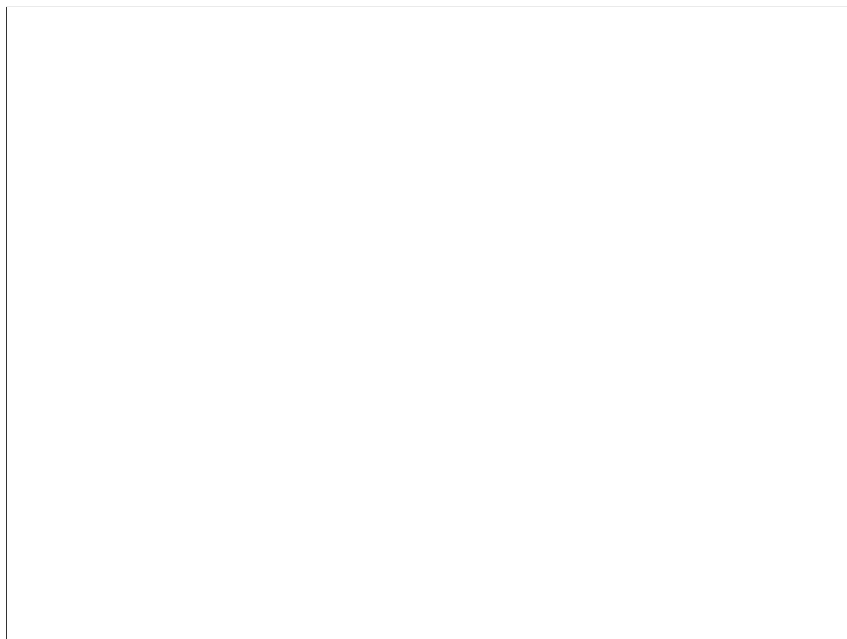
Fonte: Elaborada pelo autor a partir do *dataset* St-Data12 de Di, Gao e AlRegib (2019a).

Figura 44: Predição do *inline 390* do *dataset StData-12* com a rede DNFS com fator 64.



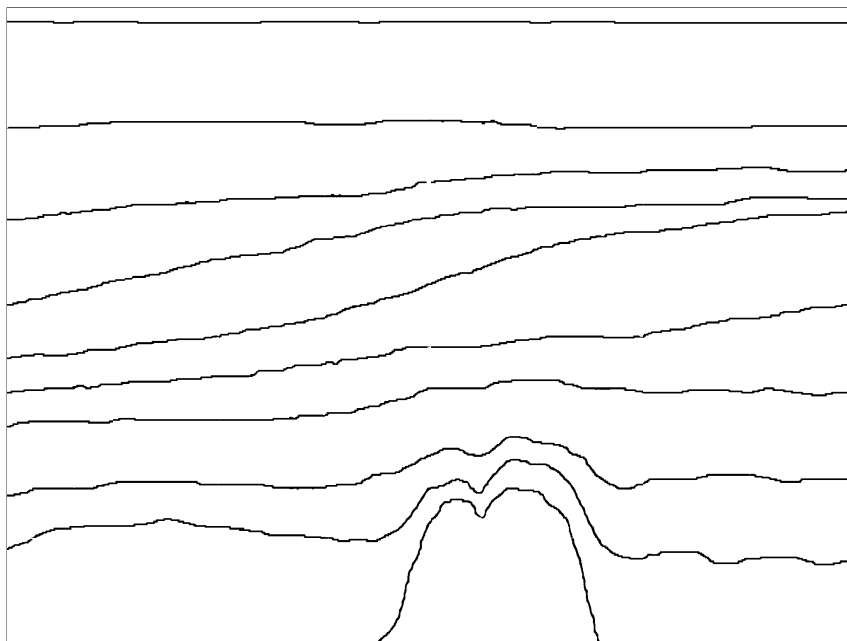
Fonte: Elaborada pelo autor a partir do *dataset St-Data12* de Di, Gao e AlRegib (2019a).

Figura 45: Predição do *inline 390* do *dataset StData-12* com a rede StNet.



Fonte: Elaborada pelo autor a partir do *dataset St-Data12* de Di, Gao e AlRegib (2019a).

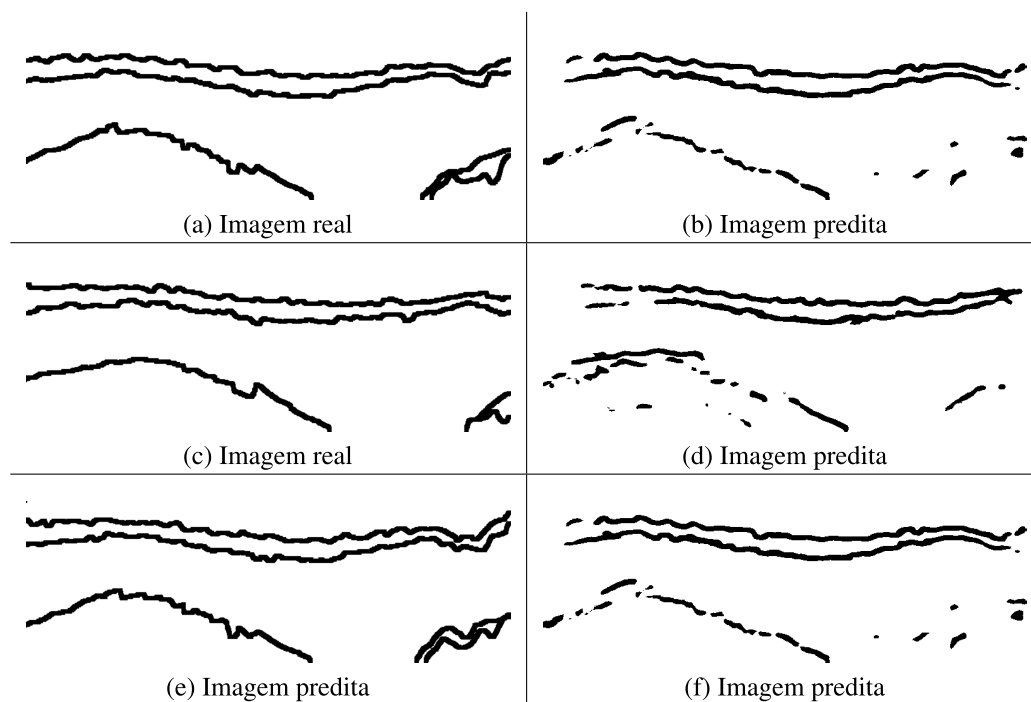
Figura 46: Predição do *inline* 390 do *dataset* StData-12 com a rede U-Net.



Fonte: Elaborada pelo autor a partir do *dataset* St-Data12 de Di, Gao e AlRegib (2019a).

b]

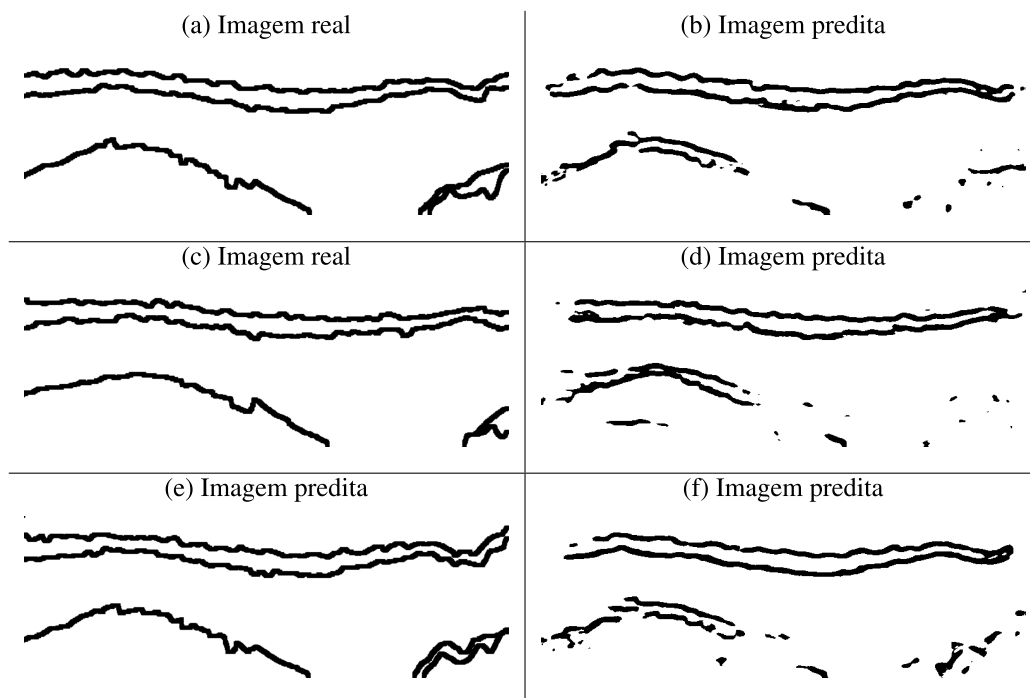
Figura 47: Predições de um *inline* presente no *dataset* *Facies-Mark* feitas pela DNFS com fator 4.



Fonte: Elaborada pelo autor a partir do *dataset* *Facies-Mark* de Alfarraj (2019).

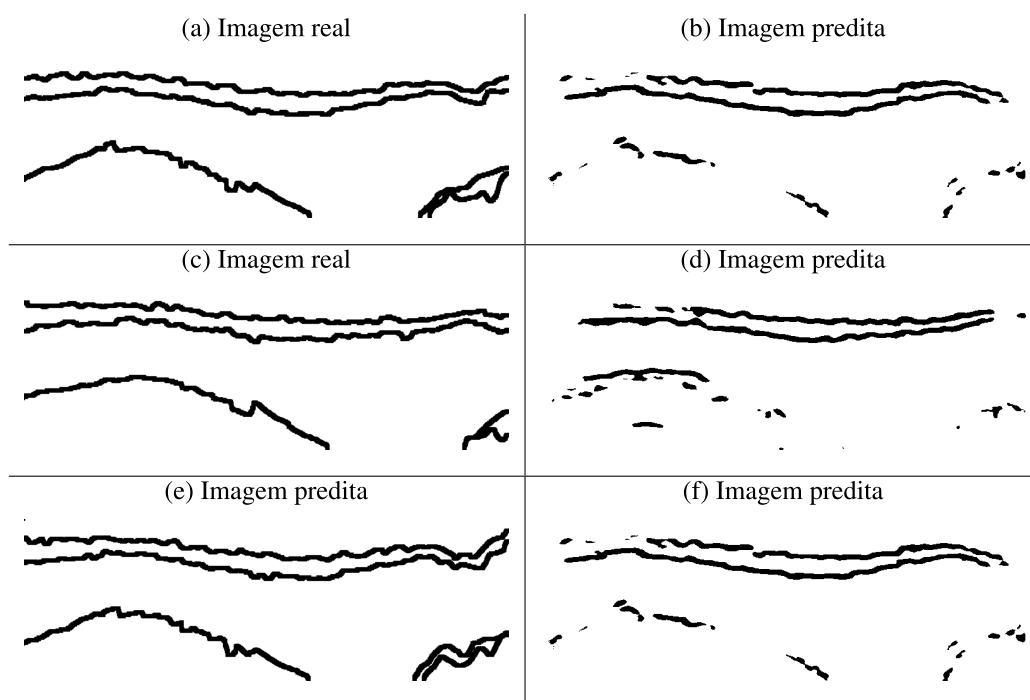
As Figuras 47, 48, 49, 50, 51, 52 e 53 exibem alguns exemplos de predições das variantes das redes DNFS e as predições da StNet e U-Net para o *dataset* *Facies-Mark*. Como pode ser observado nessas figuras, a DNFS foi capaz de fornecer predições das linhas pretas que representam as interfaces entre as fácies sísmicas, mesmo com o fator 4, sendo a menor rede entre as testada para o *Facies-Mark*.

Figura 48: Predições de um *inline* presente no *dataset* Facies-Mark feitas pela DNFS com fator 8.



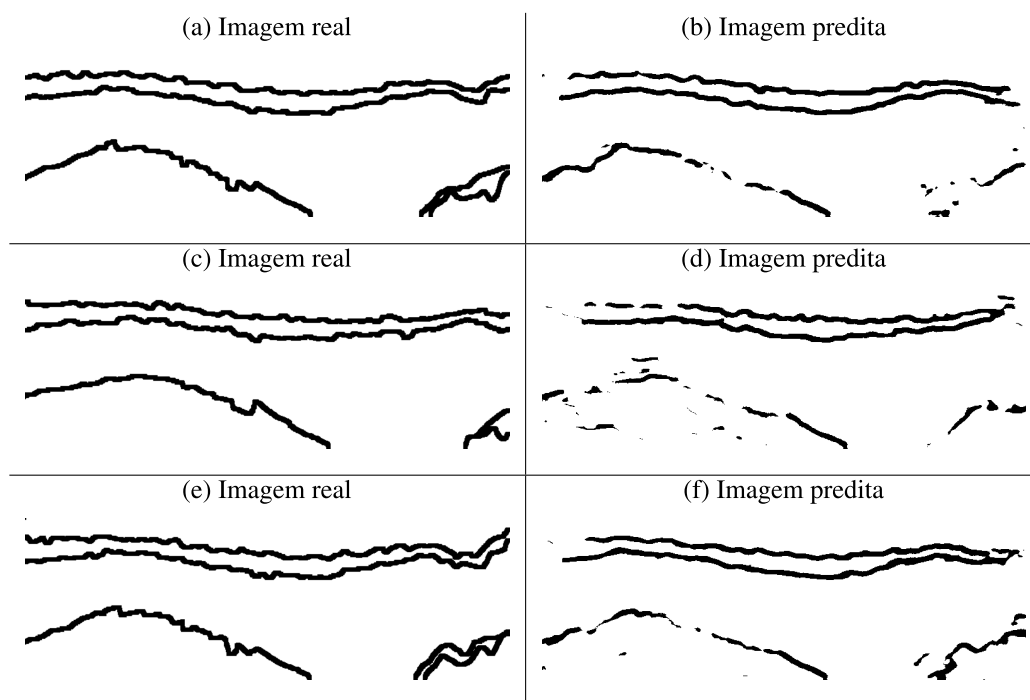
Fonte: Elaborada pelo autor a partir do *dataset* Facies-Mark de Alfarraj (2019).

Figura 49: Predições de um *inline* presente no *dataset* Facies-Mark feitas pela DNFS com fator 16.



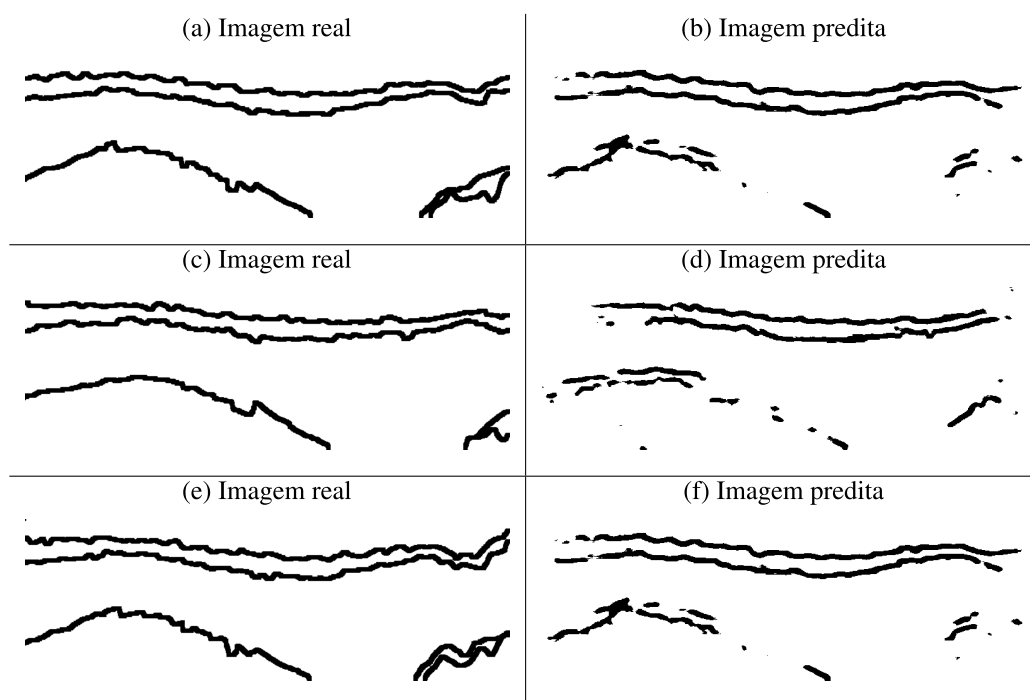
Fonte: Elaborada pelo autor a partir do *dataset* Facies-Mark de Alfarraj (2019).

Figura 50: Predições de um *inline* presente no *dataset* Facies-Mark feitas pela DNFS com fator 32.



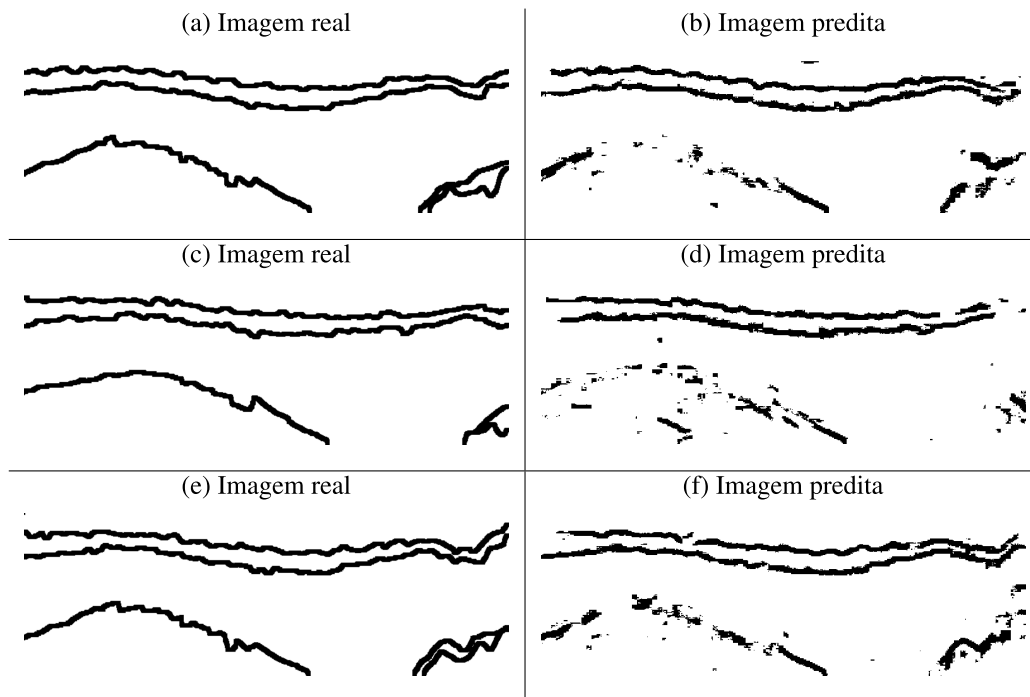
Fonte: Elaborada pelo autor a partir do *dataset* Facies-Mark de Alfarraj (2019).

Figura 51: Predições de um *inline* presente no *dataset* Facies-Mark feitas pela DNFS com fator 64.



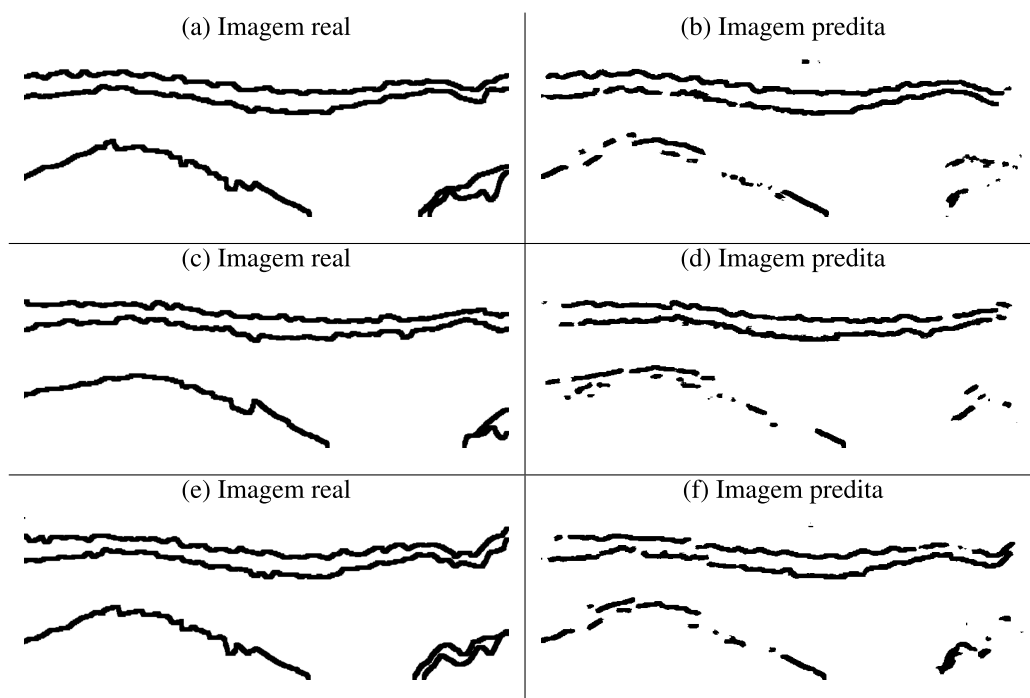
Fonte: Elaborada pelo autor a partir do *dataset* Facies-Mark de Alfarraj (2019).

Figura 52: Predições de um *inline* presente no *dataset* Facies-Mark feitas pela rede StNet.



Fonte: Elaborada pelo autor a partir do *dataset* Facies-Mark de Alfarraj (2019).

Figura 53: Predições de um *inline* presente no *dataset* Facies-Mark feitas pela rede U-Net.



Fonte: Elaborada pelo autor a partir do *dataset* Facies-Mark de Alfarraj (2019).

5.4 Considerações

Os experimentos descrito neste capítulo puderam demonstrar que o treinamento de redes neurais do tipo *encoder-decoder* a partir de pequenos segmentos de imagens sísmicas é viável e capaz de fornecer predições de linhas que identificam as interfaces entre fácies sísmicas. A utilização das redes DNFS, StNet e U-Net, a função de métrica *Jaccard distance* e a combinação das funções de perda *cross-entropy* e *Jaccard loss* foram capazes de auxiliar no treinamento e na mensuração das imagens preditas por essas redes, onde múltiplas fácies sísmicas foram segmentadas simultaneamente com menos de 30 minutos de treinamento das redes neurais. Com esse experimento, observou-se que um adequado uso dos parâmetros de redes *encoder-decoder* é importante ao ponto de que, como sugere o experimento realizado, somente o aumento da quantidade de parâmetros parece não ter capacidade de aumentar a acurácia das predições realizadas pela rede. Uma outra característica importante notada nesse experimento foi que o modo como é feita a transição na quantidade de filtros entre as camadas de ser levado em consideração, pois a diminuição acentuada no número de filtros, poderia gerar perda de informação latente do dado de entrada, conseqüentemente, comprometendo os resultados preditos pela rede.

Ao final, esse experimento pôde demonstrar variantes da rede DNFS, um modo de adaptação de *datasets* para uso de segmentação binária através de redes neurais utilizando dados sísmicos e a combinação de funções de perda otimizadas para a identificação de múltiplas fácies sísmicas através das interfaces entre elas.

6 CONCLUSÕES E TRABALHOS FUTUROS

A interpretação sísmica é uma parte inicial importante da exploração de petróleo em mar, pois possui um custo menor para a aquisição de dados comparado ao se perfurar poços em alto-mar. Para realizar a interpretação é necessário haver um especialista e entendimento da área de estudo. Com a intenção de gerar subsídios para apoiar o intérprete de dado sísmico, esse trabalho abordou a segmentação de objetos em dados sísmicos através do uso de redes convolucionais profundas utilizando a arquitetura *encoder-decoder*. Foram avaliadas três arquiteturas de redes neurais com essa arquitetura: DNFS, StNet e U-Net. Sendo a U-Net utilizada na segmentação binária de dados médicos, a StNet na segmentação de fácies sísmicas coloridas e a DNFS para a segmentação binária de linhas que representam as interfaces entre fácies sísmicas.

As informações utilizados no experimento foram compostas por seções sísmicas extraídas de dois *datasets*: *StData-12* e *Facies-Mark*. O *StData-12*, é um volume sísmico que contém seções sísmicas *inlines* interpretadas e o *Facies-Mark* é um volume destinado ao treinamento de redes convolucionais profundas para a segmentação de múltiplas fácies sísmicas. Durante o processo de avaliação a quantidade de filtros das camadas da DNFS foi modificada. O objetivo foi verificar quais as configurações mínimas para tal rede a fim de que a mesma fosse treinada em poucos minutos e ainda assim fornecesse predição relativas as linhas representantes das interfaces entre fácies sísmicas. Foi aferido que o uso da combinação das funções *cross-entropy* e *Jaccard loss* para o cálculo da perda e *Jaccard index* para a métrica de avaliação forneceram resultados satisfatórios para as configurações de treinamento da DNFS. Mesmo com a exceção dos casos onde uma determinada configuração para a quantidade de filtros, não forneceu um predição de linhas alguma. No decorrer do desenvolvimento do experimento se notou que o uso de *skip-connections* foi importante para o treinamento da rede, porém, sua utilização deve ser feita com cuidado, pois o seu uso excessivo pode criar ruídos nas camadas mais profundas das redes. A sua aplicação, sendo baseada no fim dos blocos de convolução com o início dos de convolução transposta parece ser a mais adequada para redes otimizadas como a DNFS. Além dos *skip-connections*, a transição na quantidade de filtros em camadas sucessivas deve ser levada em conta de modo cuidadoso. Na rede StNet a transição ocorre acentuadamente, diferentemente das redes DNFS e U-Net. Isso impacta na capacidade da rede de aprender, pois muitas informações podem ser perdidas durante a transição entre as camadas com mudanças acentuadas nas quantidades de filtros.

A aplicação de segmentação de redes neurais sobre dados sísmicos deve levar em conta que as regiões a serem segmentadas não possuem distinções claras de contorno como imagens como prédios e animais, as quais são facilmente distinguíveis por humanos. Dados sísmicos não possuem essas características e dependem fortemente das organizações laterais de cores de pixels e da importância do objeto geológico a ser segmentado, o que configura um desafio a mais para o trabalho de segmentação.

Ao final, pode-se verificar que a aplicação de uma rede convolucional profunda otimizada

para a segmentação binária de fácies sísmicas poderia ser utilizada no fluxo de trabalho diário de um intérprete de dados sísmicos.

6.1 Limitações

Durante o processo de desenvolvimento desse trabalho foram observadas as seguintes limitações:

- Somente se comparou os resultados da rede proposta (DNFS) com as redes StNet e U-Net, o que representa uma pequena parte das arquiteturas de redes neurais existentes.
- Pequena quantidade de dados utilizados para o treinamento, assim, generalizar as predições para áreas diferentes da utilizada para o treinamento se torna um ponto de posterior investigação.
- Utilização de somente dois *datasets* para comparação das predições realizadas através da segmentação binária.
- Limitação de hardware, GPU e memória RAM, para treino das redes neurais o que nos levou a usar *batch sizes* de tamanhos diferentes para cada tipo de rede neural testada.
- Fácies sísmicas que se estendem por seções sísmicas em um volume sísmico, tendem a fazer a rede proposta ter previsões com uma taxa menor de acertos, uma vez que os dados de treinamento têm apenas duas dimensões.

6.2 Trabalhos futuros

Como trabalhos futuros temos os seguintes itens:

- Investigar um função de perda que se foque nos *pixels* que compõem as interfaces entre fácies sísmicas e considere um intervalo médio entre as posições preditas e as reais dos *pixels*.
- Investigar uma métrica que possa ser utilizada durante o treinamento da rede neural que forneça a porcentagem de *pixels* preditos corretos para as interfaces entre a fácies sísmicas, assim como uma outra que se foque nas áreas onde não há essas interfaces. A primeira métrica, poderia considerar um intervalo médio entre as posições preditas e as reais dos *pixels*.
- Investigar como a porcentagem de *pixels* pretos preditos corretamente pela rede DNFS poderia ser aumentada.
- Testar a performance da rede proposta contra outras arquiteturas de redes neurais.

- Implementar um *workflow* que seja utilizável pelo intérprete de dados sísmicos para que possamos ter um *feedback* dos ajustes necessários na rede neural proposta, nas funções de perda e de métricas, assim como no modo de adaptação dos dados sísmicos para a segmentação binária.
- Verificar se o uso de segmentação tridimensional poderia melhorar nossos resultados devido à relação espacial entre partes da mesma fácies sísmica distribuída entre as seções sísmicas.
- Verificar como a estrutura da DNFS poderia ser modificada para que ao invés de, somente fornecer a classificação de cada *pixel*, ela também fizesse previsões das posições espaciais de uma mesma fácies sísmica que se estende entre diversas seções sísmicas em um volume.

REFERÊNCIAS

- ABADI, M. et al. **Tensorflow - large-scale machine learning on heterogeneous systems**. Software available from tensorflow.org.
- AGGARWAL, C. **Neural networks and deep learning: a textbook**. [S.l.]: Springer International Publishing, 2018.
- ALAUDAH, Y.; ALFARRAJ, M.; ALREGIB, G. Structure label prediction using similarity-based retrieval and weakly supervised label mapping. , [S.l.], v. 84, n. 1, p. V67–V79, 2019.
- ALAUDAH, Y.; ALREGIB, G. **Landmass seismic dataset**. 2015.
- ALAUDAH, Y. et al. A machine-learning benchmark for facies classification. **Interpretation**, [S.l.], v. 7, n. 3, p. SE175–SE187, aug 2019.
- ALAUDAH, Y.; GAO, S.; ALREGIB, G. Learning to label seismic structures with deconvolution networks and weak labels. In: _____. **Seg technical program expanded abstracts 2018**. [S.l.: s.n.], 2018. p. 2121–2125.
- ALAUDAH, Y.; SOLIMAN, M.; ALREGIB, G. Facies classification with weak and strong supervision: a comparative study. In: _____. **Seg technical program expanded abstracts 2019**. [S.l.: s.n.], 2019. p. 1868–1872.
- ALFARHAN, M. et al. Multiple events detection in seismic structures using a novel u-net variant. In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (ICIP), 2020., 2020. **Anais...** IEEE, 2020.
- ALFARRAJ, G. A. Y. A. P. M. M. **Facies-mark: a machine learning benchmark for facies classification**. [S.l.]: IEEE Dataport, 2019.
- ALSADI, H. N. **Seismic hydrocarbon exploration: 2d and 3d techniques**. [S.l.]: Springer International Publishing, 2016. (Advances in Oil and Gas Exploration & Production).
- ASA, E. **"simple and fast io for segy files**. [S.l.]: GitHub, 2019.
- ASHCROFT, W. **A petroleum geologist's guide to seismic reflection**. [S.l.]: Wiley, 2011. 16 p. (A Petroleum Geologist's Guide to Seismic Reflection).
- AYAN, A. M.; YADAV, A. Introduction to seismic data. In: _____. **Atlas of structural geological interpretation from seismic images**. [S.l.]: John Wiley & Sons, Ltd, 2018. p. 1–14.
- BABASAFARI, A. A.; SAMADIAN, A. Stratigraphic oil trap detection using seismic attributes and neural network. In: SEGJ INTERNATIONAL SYMPOSIUM, KYOTO, JAPAN, 20-22 NOVEMBER 2011, 10., 2012. **Proceedings...** [S.l.: s.n.], 2012. p. 1–4.
- BADRINARAYANAN, V.; KENDALL, A.; CIPOLLA, R. Segnet: a deep convolutional encoder-decoder architecture for image segmentation. **IEEE transactions on pattern analysis and machine intelligence**, [S.l.], v. 39, n. 12, p. 2481—2495, Dec. 2017.

BARONI, L. et al. Penobscot dataset fostering machine learning development for seismic interpretation. , [S.l.], 03 2019.

BAY, H. et al. **Geological survey of canada open file 8507**. 2019.

BERMAN, M.; RANNEN, A.; BLASCHKO, M. The lovasz-softmax loss: a tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In: OF THE 10TH , 2018. **Anais...** [S.l.: s.n.], 2018. p. 4413–4421.

BOER, P.-T. de et al. A tutorial on the cross-entropy method. **Annals of Operations Research**, [S.l.], v. 134, n. 1, p. 19–67, Feb. 2005.

BOTTOU, L. Stochastic gradient descent tricks. In: NEURAL NETWORKS TRICKS OF THE TRADE: SECOND EDITION, 2012, Berlin, Heidelberg. **Anais...** Springer Berlin Heidelberg, 2012. p. 421–436.

BRAGA, A. d. P.; CARVALHO, A. C. P. d. L. F.; LUDERMIR, T. B. **Redes neurais artificiais: teoria e aplicações**. [S.l.]: LTC Editora, 2000.

CASWELL, T. A. et al. **matplotlibplotlib**. [S.l.]: Zenodo, 2020.

CHAURASIA, A.; CULURCIELLO, E. LinkNet: exploiting encoder representations for efficient semantic segmentation. In: IEEE VISUAL COMMUNICATIONS AND IMAGE PROCESSING (VCIP), 2017., 2017. **Anais...** IEEE, 2017.

CHEN, L.-C. et al. Rethinking atrous convolution for semantic image segmentation. , [S.l.], 06 2017.

CHEN, L.-C. et al. DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, [S.l.], v. 40, n. 4, p. 834–848, apr 2018.

CHEVITARESE, D. et al. Seismic facies segmentation using deep learning. , [S.l.], 2018.

CHEVITARESE, D. et al. Deep learning applied to seismic facies classification: a methodology for training. In: SAINT PETERSBURG 2018, 2018. **Anais...** EAGE Publications BV, 2018.

CHOPRA, S.; MARFURT, K. J. **Seismic attributes for prospect identification and reservoir characterization**. [S.l.]: Society of Exploration Geophysicists and European Association of Geoscientists and Engineers, 2007.

CIRESAN, D. et al. Deep neural networks segment neuronal membranes in electron microscopy images. In: PEREIRA, F. et al. (Ed.). **Advances in neural information processing systems 25**. [S.l.]: Curran Associates, Inc., 2012. p. 2843–2851.

CIVITARESE, D. et al. Semantic segmentation of seismic images. , [S.l.], 2019.

DEAN, J. et al. Large scale distributed deep networks. In: PEREIRA, F. et al. (Ed.). **Advances in neural information processing systems 25**. [S.l.]: Curran Associates, Inc., 2012. p. 1223–1231.

DI, H.; ALREGIB, G. A comparison of seismic saltbody interpretation via neural networks at sample and pattern levels. , [S.l.], v. 68, n. 2, p. 521–535, 2020.

- DI, H.; GAO, D.; ALREGIB, G. Developing a seismic texture analysis neural network for machine-aided seismic pattern recognition and classification. **Geophysical Journal International**, [S.l.], v. 218, n. 2, p. 1262–1275, 05 2019.
- DI, H.; GAO, D.; ALREGIB, G. Developing a seismic texture analysis neural network for machine-aided seismic pattern recognition and classification. **Geophysical Journal International**, [S.l.], v. 218, n. 2, p. 1262–1275, 05 2019.
- DI, H.; WANG, Z.; ALREGIB, G. Seismic fault detection from post-stack amplitude by convolutional neural networks. In: EAGE PUBLICATIONS BV, 2018. **Anais...** [S.l.: s.n.], 2018.
- DICE, L. R. Measures of the amount of ecologic association between species. **Ecology**, [S.l.], v. 26, n. 3, p. 297–302, 1945.
- DOGO, E. M. et al. A comparative analysis of gradient descent-based optimization algorithms on convolutional neural networks. In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL TECHNIQUES, ELECTRONICS AND MECHANICAL SYSTEMS (CTEMS), 2018., 2018. **Anais...** IEEE, 2018.
- DONDURUR, D. Chapter 11 - seismic migration. In: DONDURUR, D. (Ed.). **Acquisition and processing of marine seismic data**. [S.l.]: Elsevier, 2018. p. 493–547.
- DRAMSCH, J. S.; LÜTHJE, M. Deep-learning seismic facies on state-of-the-art cnn architectures. In: _____. **Seg technical program expanded abstracts 2018**. [S.l.: s.n.], 2018. p. 2036–2040.
- FIDON, L. et al. Generalised wasserstein dice score for imbalanced multi-class segmentation using holistic convolutional networks. , [S.l.], 07 2017.
- FRED, L. **Machine learning — a matemática da aprendizagem supervisionada**. (Accessed on 07/08/2019), .
- FU, J. et al. Dual attention network for scene segmentation. In: IEEE/CVF CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), 2019., 2019. **Anais...** IEEE, 2019.
- GAO, D. Integrating 3d seismic curvature and curvature gradient attributes for fracture characterization: methodologies and interpretational implications. **GEOPHYSICS**, [S.l.], v. 78, n. 2, p. O21–O31, 2013.
- GLOROT, X.; BORDES, A.; BENGIO, Y. Deep sparse rectifier neural networks. In: FOURTEENTH INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND STATISTICS, 2011, Fort Lauderdale, FL, USA. **Proceedings...** JMLR Workshop and Conference Proceedings, 2011. p. 315–323. (Proceedings of Machine Learning Research, v. 15).
- GONZALEZ, R.; WOODS, R. **Digital image processing**. [S.l.]: Pearson, 2018.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning**. [S.l.]: MIT Press, 2016. (Adaptive Computation and Machine Learning series).
- GRAUPE, D. **Principles of artificial neural networks**. 2nd. ed. River Edge, NJ, USA: World Scientific Publishing Co., Inc., 2007.

GUZZELLI, A. B.; ROISENBERG, M.; RODRIGUES, B. B. Efficient 3d semantic segmentation of seismic images using orthogonal planes 2d convolutional neural networks. In: INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS (IJCNN), 2020., 2020. **Anais...** IEEE, 2020.

GUO, J. et al. A deep supervised edge optimization algorithm for salt body segmentation. **IEEE Geoscience and Remote Sensing Letters**, [S.l.], p. 1–5, 2020.

GUPTA, H. et al. Deep learning-based automatic horizon identification from seismic data. In: SPE ANNUAL TECHNICAL CONFERENCE AND EXHIBITION, 2019. **Anais...** Society of Petroleum Engineers, 2019.

HAYKIN, S. **Redes neurais - 2ed.** [S.l.]: Bookman, 2001.

HE, K. et al. Deep residual learning for image recognition. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), 2016., 2016. **Anais...** IEEE, 2016.

HEATON, J. **Artificial intelligence for humans, volume 3: deep learning and neural networks.** [S.l.]: CreateSpace Independent Publishing Platform, 2015. (Artificial Intelligence for Humans Series).

HOCHREITER, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. **International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems**, [S.l.], v. 6, p. 107–116, 04 1998.

HONG, S. et al. Joint image clustering and labeling by matrix factorization. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, [S.l.], v. 38, n. 7, p. 1411–1424, jul 2016.

IKELLE, L.; AMUNDSEN, L. **Introduction to petroleum seismology, second edition.** [S.l.]: Society of Exploration Geophysicists, 2018. (Investigations in Geophysics Series).

IOFFE, S.; SZEGEDY, C. Batch normalization: accelerating deep network training by reducing internal covariate shift. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING - VOLUME 37, 32., 2015. **Proceedings...** JMLR.org, 2015. p. 448–456. (ICML'15).

ISLAM, M. S. ul. Using deep learning based methods to classify salt bodies in seismic images. , [S.l.], p. 104054, 2020.

KAINKARYAM, S. et al. Crowdsourcing salt model building kaggle-tgs salt identification challenge. , [S.l.], v. 2019, n. 1, p. 1–5, 2019.

KEAREY, P.; BROOKS, M.; HILL, I. **An introduction to geophysical exploration.** [S.l.]: Wiley, 2002. (Geoscience texts).

KINGMA, D.; BA, J. Adam a method for stochastic optimization. **International Conference on Learning Representations**, [S.l.], 12 2014.

KITCHENHAM, B. A.; CHARTERS, S. **Guidelines for performing systematic literature reviews in software engineering.** [S.l.]: Keele University and Durham University Joint Report, 2007. (EBSE 2007-001).

KRISENERGY. **Exploration - krisenergy 2019.** (Accessed on 07/07/2019), .

- KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. Imagenet classification with deep convolutional neural networks. **Neural Information Processing Systems**, [S.l.], v. 25, 01 2012.
- LC, G. **Colaboratory**. [S.l.]: Google, 2018.
- LEE, H. et al. Pixel-level deep segmentation: artificial intelligence quantifies muscle on computed tomography for body morphometric analysis. **Journal of Digital Imaging**, [S.l.], v. 30, n. 4, p. 487–498, Aug. 2017.
- LI, F. et al. ADDCNN: an attention-based deep dilated convolutional neural network for seismic facies analysis with interpretable spatial–spectral maps. **IEEE Transactions on Geoscience and Remote Sensing**, [S.l.], v. 59, n. 2, p. 1733–1744, feb 2021.
- LIANG, X. et al. Proposal-free network for instance-level object segmentation. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, [S.l.], v. 40, n. 12, p. 2978–2991, Dec. 2018.
- LIN, T.-Y. et al. Focal loss for dense object detection. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, [S.l.], v. 42, n. 2, p. 318–327, feb 2020.
- LIU, M. et al. Seismic facies classification using supervised convolutional neural networks and semisupervised generative adversarial networks. **Geophysics**, [S.l.], v. 85, n. 4, p. O47–O58, 06 2020.
- LIU, Z. et al. Interpretability-guided convolutional neural networks for seismic fault segmentation. In: ICASSP 2020 - 2020 IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING (ICASSP), 2020. **Anais...** IEEE, 2020.
- LONG, J.; SHELHAMER, E.; DARRELL, T. Fully convolutional networks for semantic segmentation. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), 2015., 2015. **Anais...** IEEE, 2015.
- MA, Y. et al. Automatic first arrival picking for borehole seismic data using a pixel-level network. In: SEG TECHNICAL PROGRAM EXPANDED ABSTRACTS 2019, 2019. **Anais...** [S.l.: s.n.], 2019. p. 2463–2467.
- MATTOS, A. B. et al. Enabling robust horizon picking from small training sets. **IEEE Transactions on Geoscience and Remote Sensing**, [S.l.], p. 1–8, 2020.
- MILOSAVLJEVIĆ, A. Identification of salt deposits on seismic images using deep learning method for semantic segmentation. **International Journal of Geo-Information**, [S.l.], v. 9, p. 24, 01 2020.
- MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. **Foundations of machine learning**. [S.l.]: The MIT Press, 2012.
- MORAIS, J. M. d. **Petróleo em águas profundas: uma história tecnológica da petrobras na exploração e produção offshore**. [S.l.]: Ipea, 2013.
- MOSSER, L.; OLIVEIRA, R.; STEVENTON, M. Probabilistic seismic interpretation using bayesian neural networks. In: EAGE CONFERENCE AND EXHIBITION 2019, 81., 2019. **Anais...** European Association of Geoscientists & Engineers, 2019. v. 2019, p. 1–5.

NANDA, N. C. Seismic interpretation methods. In: SEISMIC DATA INTERPRETATION AND EVALUATION FOR HYDROCARBON EXPLORATION AND PRODUCTION: A PRACTITIONER'S GUIDE, 2016, Cham. **Anais...** Springer International Publishing, 2016. p. 37–72.

NASIM, M. Q. et al. Seismic facies analysis: a deep domain adaptation approach. , [S.l.], 2020.

ONAJITE, E. **Seismic data analysis techniques in hydrocarbon exploration**. [S.l.]: Elsevier Science, 2013.

RAHMAN, M. A.; WANG, Y. Optimizing intersection-over-union in deep neural networks for image segmentation. In: ADVANCES IN VISUAL COMPUTING, 2016, Cham. **Anais...** Springer International Publishing, 2016. p. 234–244.

RANDEN, T.; SØNNELAND, L. Atlas of 3d seismic attributes. In: MATHEMATICAL METHODS AND MODELLING IN HYDROCARBON EXPLORATION AND PRODUCTION, 2005, Berlin, Heidelberg. **Anais...** Springer Berlin Heidelberg, 2005. p. 23–46.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: convolutional networks for biomedical image segmentation. In: MEDICAL IMAGE COMPUTING AND COMPUTER-ASSISTED INTERVENTION – MICCAI 2015, 2015, Cham. **Anais...** Springer International Publishing, 2015. p. 234–241.

SHI, Y.; WU, X.; FOMEL, S. Automatic salt-body classification using deep-convolutional neural network. In: SEG TECHNICAL PROGRAM EXPANDED ABSTRACTS 2018, 2018. **Anais...** [S.l.: s.n.], 2018. p. 1971–1975.

SHI, Y.; WU, X.; FOMEL, S. Saltseg: automatic 3d salt segmentation using a deep convolutional neural network. **Interpretation**, [S.l.], v. 7, n. 3, p. SE113–SE122, 2019.

SILVA, I. N. da et al. **Artificial neural networks: a practical course**. [S.l.]: Springer International Publishing, 2016.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **arXiv 1409.1556**, [S.l.], 09 2014.

SOLOMON, C.; BRECKON, T. **Fundamentals of digital image processing: a practical approach with examples in matlab**. [S.l.]: Wiley, 2011.

SRIVASTAVA, N. et al. Dropout: a simple way to prevent neural networks from overfitting. **Journal of Machine Learning Research**, [S.l.], v. 15, p. 1929–1958, 06 2014.

SRIVASTAVA, R. K.; GREFF, K.; SCHMIDHUBER, J. Training very deep networks. In: INTERNATIONAL CONFERENCE ON NEURAL INFORMATION PROCESSING SYSTEMS - VOLUME 2, 28., 2015, Cambridge, MA, USA. **Proceedings...** MIT Press, 2015. p. 2377–2385. (NIPS' 15).

SULTANA, F.; SUFIAN, A.; DUTTA, P. Evolution of image segmentation using deep convolutional neural network: a survey. **Knowledge-Based Systems**, [S.l.], v. 201-202, p. 106062, 2020.

SUN, B.; SAENKO, K. Deep coral: correlation alignment for deep domain adaptation. In: **COMPUTER VISION – ECCV 2016 WORKSHOPS, 2016**, Cham. **Anais...** Springer International Publishing, 2016. p. 443–450.

TERRANUBIS. **Terranubis**. 2020.

THOMAS, J. E. **Fundamentos de engenharia de petróleo**. [S.l.]: Interciência, 2004.

TSCHANNEN, V. et al. Extracting horizon surfaces from 3D seismic data using deep learning. **Geophysics**, [S.l.], v. 85, n. 3, p. N17–N26, 04 2020.

W; N. M.; FAICHNEY, A. K. E. **Seg-y rev 1 data exchange format**. [S.l.]: Society of Exploration Geophysicists, 2002.

WALDELAND, A.; SOLBERG, A. 3d attributes and classification of salt bodies on unlabelled datasets. , [S.l.], v. 2016, n. 1, p. 1–5, 2016.

WALDELAND, A. U. et al. Convolutional neural networks for automated seismic interpretation. **The Leading Edge**, [S.l.], v. 37, n. 7, p. 529–537, 2018.

WANG, W.; YANG, F.; MA, J. Automatic salt detection with machine learning. , [S.l.], v. 2018, n. 1, p. 1–5, 2018.

WANI, M. et al. **Advances in deep learning**. [S.l.]: Springer Singapore, 2019. (Studies in Big Data).

WEN, J. et al. Systematic literature review of machine learning based software development effort estimation models. **Information and Software Technology**, [S.l.], v. 54, n. 1, p. 41–59, 2012.

WU, X. et al. FaultNet3d: predicting fault probabilities, strikes, and dips with a single convolutional neural network. **IEEE Transactions on Geoscience and Remote Sensing**, [S.l.], v. 57, n. 11, p. 9138–9155, nov 2019.

WU, X. et al. Faultseg3d: using synthetic data sets to train an end-to-end convolutional neural network for 3d seismic fault segmentation. **Geophysics**, [S.l.], v. 84, p. IM35–IM45, 02 2019.

WYOMING, S. G. S. **What is geophysics - sercel**. (Accessed on 07/06/2019), .

XU, P.; LU, W.; WANG, B. A semi-supervised learning framework for gas chimney detection based on sparse autoencoder and TSVM. **Journal of Geophysics and Engineering**, [S.l.], v. 16, n. 1, p. 52–61, 01 2019.

YANG, M. et al. DenseASPP for semantic segmentation in street scenes. In: **IEEE/CVF CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2018.**, 2018. **Anais...** IEEE, 2018.

YUAN, S. et al. Geosteering phase attributes: a new detector for the discontinuities of seismic images. **IEEE Geoscience and Remote Sensing Letters**, [S.l.], v. 16, n. 1, p. 145–149, 2019.

YUAN, Y.; WANG, J. Ocnet: object context network for scene parsing. **ArXiv**, [S.l.], v. abs/1809.00916, 2018.

ZENG, Y.; JIANG, K.; CHEN, J. Automatic seismic salt interpretation with deep convolutional neural networks. In: ICISDM 2019: PROCEEDINGS OF THE 2019 3RD INTERNATIONAL CONFERENCE ON INFORMATION SYSTEM AND DATA MINING, 2019, New York, NY, USA. **Anais...** Association for Computing Machinery, 2019. p. 16–20. (ICISDM 2019).

ZHANG, H. et al. Context encoding for semantic segmentation. In: IEEE/CVF CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2018., 2018. **Anais...** IEEE, 2018.

ZHANG, H. et al. Automatic seismic facies interpretation based on an enhanced encoder-decoder structure. In: _____. **Seg technical program expanded abstracts 2019**. [S.l.: s.n.], 2019. p. 2408–2412.

ZHANG, H. et al. Automatic seismic facies interpretation using supervised deep learning. **GEOPHYSICS**, [S.l.], v. 86, n. 1, p. IM15–IM33, 2021.

ZHAO, H. et al. Pyramid scene parsing network. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), 2017., 2017. **Anais...** [S.l.: s.n.], 2017. p. 6230–6239.

ZHAO, T.; MUKHOPADHYAY, P. A fault-detection workflow using deep learning and image processing. In: _____. **Seg technical program expanded abstracts 2018**. [S.l.: s.n.], 2018. p. 1966–1970.

ZHOU, D. et al. IoU loss for 2d/3d object detection. In: INTERNATIONAL CONFERENCE ON 3D VISION (3DV), 2019., 2019. **Anais...** IEEE, 2019.

APÊNDICE A – INFORMAÇÕES COMPLEMENTARES

As informações que nos levaram a escolha dos valores de *dropout* e *learning rate* estão apresentadas nas Seções A.1 e A.2 que fazem parte deste apêndice. O *dataset* utilizado nos testes que nos forneceram os valores destas Seções foi o StData-12 apresentado por DI; GAO; ALREGIB (2019b).

A.1 Valores de *dropout*

As Tabelas 17, 18 e 19 contêm as redes DNFS (e suas variantes), StNet e U-Net, os valores de porcentagem utilizados com a técnica de *dropout*, a quantidade de *pixels* pretos preditos corretamente e o índice IoU atingido nas predições das redes. O valor de 10% para o *dropout* foi escolhido para uso neste trabalho pois, foi o que forneceu a maior média de *pixels* pretos.

Tabela 17: Variação do valor de *dropout* e índices de *pixels* pretos preditos corretamente. A célula da tabela com asterisco (*) representa a soma dos tempos de treinamento das redes. Em negrito está a melhor média de *pixels* pretos em relação ao *dropout*. Parte 1 de 3.

Rede	Fator	<i>Dropout</i> (%)	<i>Pixels</i> pretos (%)	IoU	Treinamento (H:mm:ss)
DNFS	4	0	0	0,9927081905732900	0:02:51
DNFS	8	0	58,35289255	0,9908339384369970	0:02:29
DNFS	16	0	51,73993869	0,9904049534367090	0:02:38
DNFS	32	0	56,8635399	0,9911975060492510	0:06:02
DNFS	64	0	59,57193989	0,9909322235797280	0:16:49
StNet	–	0	0	0,9927081905732900	0:01:05
Unet	–	0	0	0,9927081905732900	0:28:25
Média	–		32,36118729	0,9916418847460790	1:00:19*
DNFS	4	10	0	0,9927081905732900	0:02:35
DNFS	8	10	50,99955306	0,9894982455405490	0:05:17
DNFS	16	10	58,80255187	0,9908402069016500	0:08:16
DNFS	32	10	60,23108144	0,9908550099072620	0:09:58
DNFS	64	10	61,53802091	0,9912682410144940	0:36:22
StNet	–	10	0	0,9927081905732900	0:02:32
Unet	–	10	63,41947015	0,9917846866663470	0:49:29*
Média	–	–	42,14152535	0,9913803958824120	1:54:29*
DNFS	4	20	0	0,9927081905732900	0:03:53
DNFS	8	20	56,92098914	0,9907222427167810	0:05:07
DNFS	16	20	47,4314219	0,9905709755869860	0:04:44
DNFS	32	20	59,90617589	0,9912409539205210	0:12:18
DNFS	64	20	61,27294048	0,9912504845626660	0:31:49
StNet	–	20	0	0,9927081905732900	0:02:02
Unet	–	20	61,49185318	0,9912600004779060	1:02:11
Média	–	–	41,00334008	0,9914944340587770	2:02:04*

Fonte: Elaborada pelo autor.

Tabela 18: Variação do valor de *dropout* e índices de *pixels* pretos preditos corretamente. A célula da tabela com asterisco (*) representa a soma dos tempos de treinamento das redes. Parte 2 de 3.

Rede	Fator	<i>Dropout</i> (%)	<i>Pixels</i> pretos (%)	IoU	Treinamento (H:mm:ss)
DNFS	4	30	0	0,9927081905732900	0:01:50
DNFS	8	30	0	0,9927081905732900	0:02:31
DNFS	16	30	0,323807372	0,9915128035704380	0:02:10
DNFS	32	30	0,026042232	0,9926238239901290	0:01:00
DNFS	64	30	0,82106314	0,9872036779323410	0:50:02
StNet	–	30	0	0,9927081905732900	0:20:02
Unet	–	30	52,45215311	0,9907244640801460	0:38:09
Média	–	–	7,66043798	0,9914556201847030	1:07:43*
DNFS	4	40	0	0,9927081905732900	0:04:01
DNFS	8	40	20,42097363	0,9887191410936730	0:02:41
DNFS	16	40	51,55916082	0,9895472386095200	0:04:41
DNFS	32	40	61,01778121	0,9912241938364730	0:06:54
DNFS	64	40	61,36165025	0,9914520583375280	0:23:30
StNet	–	40	0	0,9927081905732900	0:01:26
Unet	–	40	56,35946062	0,9894203444045330	0:38:11
Média	–	–	35,81700379	0,9908256224897580	1:21:24*
DNFS	4	50	0	0,9927081905732900	0:02:29
DNFS	8	50	10,6292601	0,9898194626076760	0:04:27
DNFS	16	50	42,10229702	0,9903715577502930	0:03:49
DNFS	32	50	60,72160722	0,9912836530150420	0:08:09
DNFS	64	50	1,43511859	0,9404039280332520	0:03:36
StNet	–	50	0	0,9927081905732900	0:02:05
Unet	–	50	0	0,9927081905732900	0:31:48
Média	–	–	16,41261185	0,9842861675894480	0:56:22*
DNFS	4	60	0	0,9927081905732900	0:02:07
DNFS	8	60	0,021744803	0,9926381817983850	0:01:45
DNFS	16	60	0,308522946	0,9917142071377690	0:01:57
DNFS	32	60	55,2395048	0,9905763478763270	0:15:38
DNFS	64	60	0	0,9927081905732900	0:02:42
StNet	–	60	0	0,9927081905732900	0:03:57
Unet	–	60	57,9621071	0,9915493779869390	0:58:02
Média	–	–	16,21883995	0,9920860980741840	1:26:08*
DNFS	4	70	0	0,9927081905732900	0:02:49
DNFS	8	70	0	0,9927081905732900	0:02:37
DNFS	16	70	0,004386638	0,9927006083867030	0:01:43
DNFS	32	70	0,844477931	0,9898566865582870	0:01:24
DNFS	64	70	0	0,9927081905732900	0:22:26
StNet	–	70	0,004387312	0,9927017373080130	0:02:05
Unet	–	70	0	0,9927081905732900	0:05:29
Média	–	–	0,12189313	0,9922988277923090	0:38:31*

Fonte: Elaborada pelo autor.

Tabela 19: Variação do valor de *dropout* e índices de *pixels* pretos preditos corretamente. A célula da tabela com asterisco (*) representa a soma dos tempos de treinamento das redes. Parte 3 de 3.

Rede	Fator	<i>Dropout</i> (%)	<i>Pixels</i> pretos (%)	IoU	Treinamento (H:mm:ss)
DNFS	4	80	0	0,9927081905732900	0:01:45
DNFS	8	80	0,013111888	0,9926743152404470	0:02:11
DNFS	16	80	0,664124954	0,9896222669635520	0:03:14
DNFS	32	80	0	0,9926901277658580	0:03:13
DNFS	64	80	2,30218281	0,9926604728864100	0:11:59
StNet	–	80	0	0,9927081905732900	0:05:00
Unet	–	80	7,404776002	0,9924820367178100	1:03:36
Média	–	–	1,48345652	0,9922208001029510	1:30:58*
DNFS	4	90	0	0,9927081905732900	0:03:29
DNFS	8	90	0	0,9927081905732900	0:00:57
DNFS	16	90	2,859331942	0,0502681519254097	0:01:08
DNFS	32	90	0,364305428	0,9913177682085240	0:02:46
DNFS	64	90	2,883995846	0,0598237819866726	0:02:40
StNet	–	90	0	0,9927081905732900	0:05:46
Unet	–	90	0	0,9927081905732900	0:50:54
Média	–	–	0,87251903	0,7246060663448240	1:07:40*

Fonte: Elaborada pelo autor.

A.2 Valores de *learning rate*

A Tabela 20 contém as redes DNFS (e suas variantes), StNet e U-Net, os valores de *learning rate* utilizados para a seleção do valor inicial do mesmo, a quantidade de *pixels* pretos preditos corretamente e o índice IoU atingido nas predições das redes. O valor de 0,001% para o *learning rate* foi escolhido para uso neste trabalho pois, foi o que forneceu a maior média de *pixels* pretos.

Tabela 20: Variação do valor do *learning rate* e índices de *pixels* pretos preditos corretamente. A célula da tabela com asterisco (*) representa a soma dos tempos de treinamento das redes.

Rede	Fator	<i>Learning rate</i>	<i>Pixels</i> pretos (%)	IoU	Treinamento (H:mm:ss)
DNFS	4	0,1	0,0	0,99270	0:00:50
DNFS	8	0,1	0,0	0,99270	0:00:39
DNFS	16	0,1	0,0	0,99270	0:00:32
DNFS	32	0,1	0,0	0,99270	0:00:55
DNFS	64	0,1	0,0	0,99270	0:02:25
StNet	–	0,1	0,0	0,99270	0:00:47
Unet	–	0,1	2,85427	0,99979	0:02:19
Média	–	–	0,40775	0,99979	0:08:30*
DNFS	4	0,01	0,0	0,99270	0:00:49
DNFS	8	0,01	0,0	0,99270	0:00:40
DNFS	16	0,01	0,00439	0,99270	0:01:15
DNFS	32	0,01	0,0	0,99270	0:02:30
DNFS	64	0,01	2,85647	0,99959	0:03:24
StNet	–	0,01	0,54865	0,99016	0:01:27
Unet	–	0,01	0,0	0,99270	0:05:01
Média	–	–	0,48707	0,99332	0:15:08*
DNFS	4	0,001	0,0	0,9927	00:02:35
DNFS	8	0,001	51,00	0,9895	00:05:16
DNFS	16	0,001	58,80	0,9908	00:08:16
DNFS	32	0,001	60,23	0,9910	00:09:58
DNFS	64	0,001	61,54	0,9913	00:36:22
StNet	–	0,001	0,0	0,9927	00:02:31
Unet	–	0,001	63,42	0,9918	00:49:29
Média	–	–	42,14	0,9914	01:54:29*

Fonte: Elaborada pelo autor.