

# Relatório - Módulo 8

## Inteligencia Artificial Aplicada

Oliveira, V.S.P.P.<sup>1</sup>; Astolfo, J.P.I.,<sup>2</sup>; Fernandes, G.G.D.,<sup>3</sup>

<sup>1</sup>Departamento de Química e Física, Universidade Federal do Espírito Santo, Alegre, ES, Brasil

<sup>2</sup>Departamento de Física, Universidade Federal de São Carlos, São Carlos, SP, Brasil

<sup>3</sup>Departamento de Física, Instituto Superior Tecnico, Lisboa, LIS, Portugal

### Resumo

*O mapeamento das camadas do assoalho oceânico é um desafio atual para a indústria petrolífera. Os métodos de solução atuais fazem o mapeamento pelo método sísmico e inversão de ondas, com tratamento complexo e computacionalmente custoso. A introdução de redes neurais artificiais, UNet, para a predição dos modelos de velocidade a partir de tiros sísmicos refletidos pelo fundo oceânico é promissora para a otimização deste processo. Neste trabalho duas arquiteturas de redes neurais são validadas para a inversão de modelos de velocidade e comparadas quanto à estabilidade de métricas como loss function e coeficiente de similaridade e à própria diferença entre os modelos preditos e os reais. De fato as redes neurais se mostram promissoras como solução para esse desafio, obtendo valores acima de 70% para o coeficiente de Sørensen-Dice.*

## 1 Introdução

O método sísmico consiste em captar ondas mecânicas, disparadas por fontes artificiais, em sensores posicionados na superfície terrestre ou marítima, no assoalho oceânico e até mesmo dentro de poços perfurados [1]. Partindo do pressuposto que a Terra é um meio acústico, cada sensor localizado, tem como principal parâmetro a velocidade de propagação da onda, que varia à medida que ultrapassa diferentes camadas da superfície. Cada disparo é registrado, contendo características pertinentes ao posicionamento do solo, que mais tarde são conver-

tidos em uma imagem das camadas geológicas em subsuperfície.

O ganho dessa técnica previne riscos exploratórios, uma vez que se possui um prévio conhecimento geológico do terreno. Largamente utilizado pelo setor petrolífero para maximizar o aproveitamento econômico e minimizar impactos ambientais, geram-se imagens sísmicas bidimensionais que são segmentadas para identificação de possíveis jazidas de petróleo. No entanto, há uma desvantagem no uso dessa técnica. Visto que para geração de um bom modelo, a partir dos tiros, exige-se um alto custo computacional, sendo essencialmente um problema

inverso mal posto e altamente sensível às condições iniciais.

Para prevenir isso, temos como alternativa o uso de métodos automáticos de segmentação, que incluem o uso de Aprendizagem Profunda (AP), mais especificamente Redes Neurais Artificiais (RNAs). Como sugere seu nome, elas são técnicas computacionais inspirados no sistema nervoso central, especializada no reconhecimento de padrões. Assim como o cérebro humano, seu aprendizado parte do uso de pesos sinápticos, que conectam os neurônios da rede, que juntos formam uma camada; à medida que informações passam pelas camadas de entrada, são distribuídas paralelamente para todos os neurônios, que juntos processam os valores, e no final um sinal é propagado na camada de saída.

Partindo desse pressuposto, tivemos como objetivo geral utilizar a RNA, UNet, para o processo de segmentação. Tendo com objetivos específicos, fazer o pré processamento das imagens sísmicas, por meio do algoritmo *deep wave*, treinar os modelos de AP com diferentes hiperparâmetros e realizar comparações estatísticas por meio de métricas, gráficos de box-plot e das imagens geradas pelo modelo em comparação com o Ground Truth (GT). O resultado foram modelos de UNet com taxas acima de 70% a métrica do coeficiente do Sørensen-Dice, demonstrando potencial da rede para o processo de identificação de petróleo nas imagens.

## 2 Referencial teórico

### 2.1 Redes neurais artificiais

RNAs são modelos computacionais inspirados no funcionamento do cérebro humano, que buscam aprender a partir de

dados de entrada e, assim, realizar tarefas específicas. Esses modelos são formados por um conjunto de neurônios interconectados, que processam informações e geram saídas com base em regras matemáticas.

A estrutura de uma RNA é composta por camadas de neurônios, sendo que cada camada pode ter um número variável de neurônios. A camada de entrada recebe os dados de entrada da rede, enquanto as camadas ocultas processam esses dados e geram novas informações. Por fim, a camada de saída produz a resposta final da rede [2].

Uma das principais vantagens das RNA é sua capacidade de aprender a partir de exemplos. Durante o treinamento da rede, são apresentados vários exemplos de entrada e a saída esperada para cada um deles. A rede ajusta seus pesos de forma a minimizar a diferença entre a saída produzida e a saída esperada. Esse processo é chamado de aprendizado por retro propagação.

Existem diversos tipos de RNAs, cada uma com sua própria arquitetura e aplicação específica. Alguns exemplos incluem (em inglês, *Convolutional Neural Network CNN*), utilizadas em tarefas de visão computacional, as redes neurais recorrentes, utilizadas em tarefas que envolvem sequências de dados, e as redes neurais de alimentação direta, utilizadas em problemas de classificação e regressão [3].

As RNAs têm sido amplamente utilizadas em diversas áreas, como reconhecimento de padrões, processamento de fala, processamento de imagens e jogos. Além disso, com o aumento do poder computacional e a disponibilidade de grandes conjuntos de dados, as redes neuronais têm se mostrado cada vez mais eficazes em tarefas complexas, como o reconhecimento

de objetos em imagens e a tradução automática de idiomas [4].

Apesar das suas vantagens, as RNAs ainda apresentam algumas limitações. Em geral, elas requerem grandes quantidades de dados de treinamento para alcançar um bom desempenho, e podem ser bastante sensíveis a ruídos e variações nos dados. Além disso, a interpretabilidade das RNAs pode ser um desafio, já que os pesos e conexões internas da rede são difíceis de serem compreendidos pelos seres humanos [5].

### 2.1.1 Rede Neural Convolucional: Unet

CNN é tipo particular de RNA que recebe conjuntos de imagens como dados de entrada, tendo como saída um rótulo; basicamente, podem ser resumidas como um conjunto de camadas de convolução e *pooling*, ao qual recebem tensores de entrada de forma (altura, largura, canais), não incluindo a dimensão do lote [6].

Neste trabalho, foi utilizado um tipo de CNN, a UNet. Desenvolvida em 2015 por Ronneberger, Fischer e Brox [7]; exemplificada pela Figura 1, ela recebe esse nome devido ao formato de U de sua arquitetura. Inicialmente, foi concebida e aplicada em imagens biomédicas, devido ao seu desempenho superior a esses tipos de imagens; e também dada a sua eficiência de obter resultados para pequenos conjuntos de dados [7]. A diferença entre uma CNN e uma UNet está pela camada de saída, uma vez que a primeira entrega um rótulo, a outra são imagens.

Outra grande diferença está pelo uso das *skip connections*. Como é observada pela Figura 1 são ligações feitas entre as camadas dos blocos codificadoras e decodificadoras, permitindo recuperar infor-

mação espacial perdida através de operações feitas nas camadas de *pooling* [8].

## 2.2 Inversão de modelos de velocidades sísmicas usando técnicas de Deep-Learning

A inversão de modelos de velocidades sísmicas é uma técnica utilizada na exploração de petróleo e gás, na qual se busca inferir a estrutura interna do subsolo a partir de dados sísmicos. O objetivo é obter um modelo de velocidades que possa ser usado para alocar poços de perfuração e identificar reservatórios de hidrocarbonetos.

As técnicas de inversão tradicionais são baseadas em métodos de otimização que requerem a definição de um modelo inicial e a minimização de uma função objetivo. No entanto, esses métodos são limitados pela presença de ruídos nos dados sísmicos e pela falta de informação sobre a estrutura geológica do subsolo [9].

Recentemente, técnicas baseadas em AP têm sido propostas para realizar a inversão de modelos de velocidades sísmicas. Essas técnicas utilizam RNAs profundas para aprender a relação entre os dados sísmicos e o modelo de velocidades. Em vez de definir explicitamente um modelo inicial e uma função objetivo, a RNA é treinada para mapear os dados sísmicos para um modelo de velocidades que produza saídas similares aos dados reais.

Uma das principais vantagens dessas técnicas é sua capacidade de lidar com dados complexos e com ruídos. As RNAs são capazes de aprender padrões complexos nos dados sísmicos e de filtrar os ruídos, produzindo modelos de velocidades mais precisos. Além disso, as RNAs po-

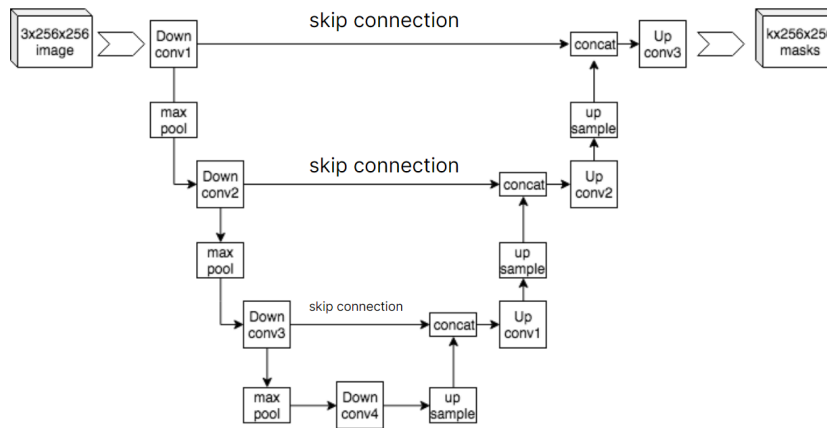


Figura 1: Exemplo de uma UNet. Dada a complexidade das RNAs utilizadas neste trabalho, temos no lugar uma exemplificação de suas arquiteturas

dem incorporar informações geológicas adicionais, como dados de poços e informações geofísicas, para produzir modelos mais precisos [10].

Uma das técnicas de AP mais utilizadas para a inversão de modelos de velocidades sísmicas é a rede neural convolucional. Essa técnica utiliza camadas convolucionais para extrair características dos dados sísmicos e camadas de regressão para produzir o modelo de velocidades. Durante o treinamento, a rede neural é ajustada para minimizar a diferença entre o modelo de velocidades produzido e o modelo real.

Outra técnica de AP utilizada na inversão de modelos de velocidades sísmicas é a rede neural generativa adversarial. Essa técnica utiliza duas redes neurais, uma geradora e outra discriminadora, que são treinadas em conjunto. A rede geradora produz modelos de velocidades sintéticos, enquanto a rede discriminadora avalia a qualidade desses modelos. Durante o treinamento, a rede geradora é ajustada para produzir modelos que enganem a rede discriminadora e sejam similares aos modelos reais.

As técnicas de inversão de modelos de velocidades sísmicas baseadas em AP

têm mostrado resultados promissores em estudos experimentais e em aplicações práticas. No entanto, ainda há desafios a serem enfrentados, como a interpretabilidade dos modelos produzidos pelas RNAs e a necessidade de grandes conjuntos de dados para o treinamento das redes [11].

Em resumo, as RNAs são uma poderosa ferramenta para a resolução de problemas complexos de aprendizado de máquina. Com sua capacidade de aprender a partir de exemplos e sua adaptabilidade a diferentes tipos de dados, elas têm se mostrado cada vez mais relevantes em diversas áreas. No entanto, ainda há muito a ser explorado em relação à interpretabilidade e às limitações desses modelos.

### 3 Materiais e métodos

Para o treinamento da RNA, foram apresentados os dados de entrada, sendo os tiros gerados pelos sensores, e como dados de saída, os modelos. Os dados sísmico foram gerados com o *DeepWave*, um pacote de simulação de propagação de ondas, permitindo realizar a modelação para a frente e a retropropagação para calcular o gradiente das saídas em rela-

ção às entradas (e assim realizar a inversão/otimização) [12].

Em seguida, foi feita a seleção dos hiperparâmetros dos dois modelos de UNet utilizadas. A grande diferença das duas redes está para suas últimas camada, onde uma delas não possui um *skip connection* entre sua primeira e última camada. Dessa forma, nomeamos como UNet e UnetMod, respectivamente.

Toda a documentação foi construída a partir da linguagem de programação Python, desenvolvido na plataforma Google Colaboratory, pela interfaces TensorFlow.Keras para o desenvolvimento dos modelos de IA.

### 3.1 Conjuntos de dados

O conjunto de dados para o treinamento e validação da rede é composto de tiros sísmicos simulados a partir de modelos de velocidade, também simulados. O pacote *DeepWave* foi utilizado para a geração dos tiros sísmicos. Cada conjunto de tiros é um *input* para a RNA. De forma a avaliar a influência da extensão do *dataset* para a eficiência da rede, dois conjuntos de dados foram criados, o primeiro com 300 elementos simulados e o segundo com 1000 simulações. Todos os modelos de velocidades e tiros sísmicos gerados são um *grid* de dados de dimensão 128 x 128.

#### 3.1.1 Gerador dos modelos

Os modelos de velocidades são gerados considerando diferentes características geológicas, como falhas, processos de erosão e quebras e o intervalo de velocidades mínimas e máximas nos diferentes meios de propagação. Dois conjuntos de modelos de velocidades são criados. O primeiro conjunto se baseia em

modelos simples, sem defeitos geológicos, com menos elementos simulados e foi considerado no escopo deste trabalho como o conjunto simples. O segundo conjunto inclui defeitos geológicos, possuem mais elementos simulados e apresentam maior complexidade, sendo considerado o conjunto complexo. A Figura 2 apresenta amostras dos modelos gerados, contendo características geológicas mais complexas.

#### 3.1.2 Gerador dos tiros

Os tiros sísmicos utilizados de *input* para as redes utilizam o pacote *DeepWave*, que implementa propagadores de onda como módulos do PyTorch. Suas funções permitem a modelagem das ondas e a retropropagação na RNA para calcular o gradiente dos *outputs* da rede em relação aos *inputs*. Diferentes parâmetros dos tiros são configuráveis nesse pacote, como a disposição dos receptores de onda simulados (número, posição, espaçamento, dentre outros parâmetros) e frequência das ondas.

### 3.2 Google Colabotory, TensorFlow e Supercomputador

O presente projeto foi desenvolvido inicialmente no *Google Colaboratory*. Por meio dessa plataforma é possível executar códigos em Python diretamente dos servidores da nuvem da Google, onde uma das suas grandes vantagens está no uso de suas Unidades de Processamentos Gráficos (Graphics Processing Unit - GPU). Em combinação, para o desenvolvimento das RNAs, usamos a interface Keras, que é um API de alto nível construído independentemente com base no

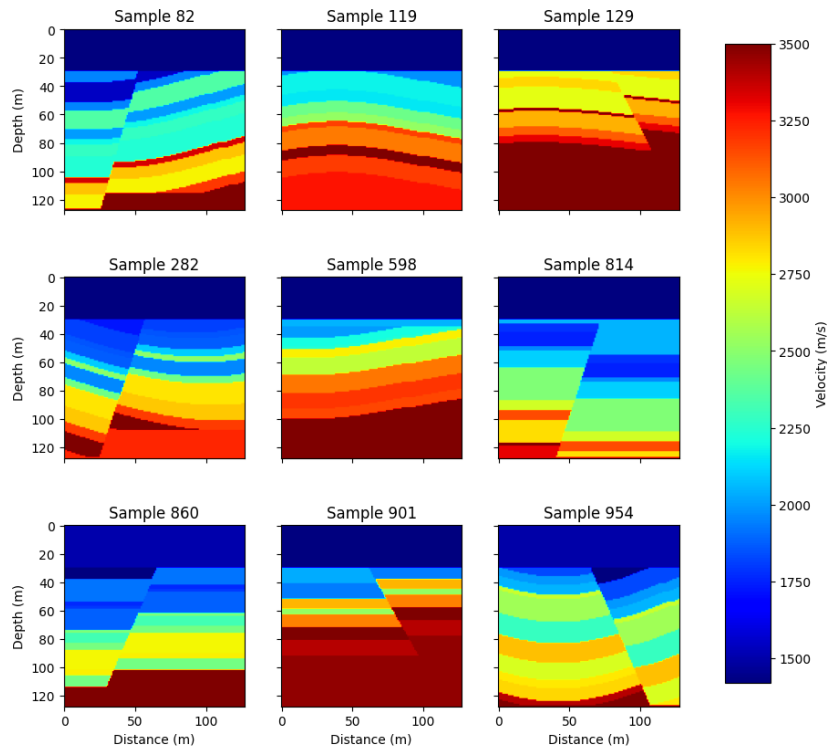


Figura 2: Amostras de modelos de velocidades gerados.

TensorFlow; uma biblioteca capaz de dividir o treinamento em vários pedaços e executá-los em paralelo em várias GPUs, podendo treinar redes neurais colossais em gigantescos conjuntos de treinamento dividindo os cálculos por centenas de servidores em um período de tempo razoável [13].

### 3.3 Treinamento da Rede Neural Artificial

Para o treinamento da RNA, foi utilizado a técnica de validação cruzada, aplicada para avaliar a capacidade de generalização do modelo em diferentes contextos; tendo como métrica de avaliação o Mean Square Error (MSE) que calcula a diferença média ao quadrado entre os valores observados e preditos. Basicamente, separou-se o conjunto em 10 lotes de forma aleatória, mantendo a porcentagem

de 64% para o treinamento e 16% para validação. Partindo desse conjunto, foi feita 10 interações de lotes aleatórias. O resultado é a média simples do resultado da métrica de todas as interações.

Com relação a etapa de teste, foram utilizados os 20% restante. Encarregado de prever novos resultados e, consequentemente, testar o desempenho da rede para dados novos, utilizou-se como métrica o Coeficiente Sørensen–Dice (DSC), criada para avaliar a taxa de similaridade entre duas imagens. Sua escolha se deu por sua intuitiva visualização, visto que a própria entrega valores entre 0.0 e 1.0, podendo ser facilmente convertidas para uma porcentagem.

Com o intuito de otimizar o treinamento, foi feita uma escolha prévia dos hiperparâmetros. Inicialmente, não houve uma preocupação direta com eles, visto que usamos funções de retorno de cha-

mada, responsáveis por monitorar os treinamentos; deu-se um maior enfoque no tamanho do lote, que consiste em pequenos conjuntos de amostras que são processados simultaneamente pelo modelo, com o intuito de facilitar o alocamento da memória da GPU [6]. Empiricamente, foi escolhido como 8 o tamanho do lote, para ambas as arquiteturas.

### 3.3.1 Análise estatística

Para uma visualização lúdica da métrica DSC, foi utilizada gráficos de box-plot, com o intuito de avaliar a estabilidade dos modelos. como complementação, realizou-se uma comparação visual do GT e as imagens preditas, verificando possíveis vícios da predição, como a presença de artefatos e outros sinais espúrios.

## 4 Resultados e discussões

A Figura 3 apresenta a curva das funções *loss* de cada arquitetura treinada neste trabalho. É notável que a UNet modificada apresenta a curva de validação mais estável em torno da curva de treinamento, quando comparada com a UNet não modificada. O comportamento das curvas de validação demonstra que a solução não apresenta problemas com *overfitting* ou *underfitting*. Ambas as redes levam aproximadamente o mesmo tempo para que suas funções *loss* deixem de apresentar variações significativas.

É interessante notar que os modelos preditos pela UNet modificada, quando comparados com os modelos reais, apresentam um artefato circular posicionado sempre na mesma região superior direita das imagens. A Figura 4 apresenta a comparação dos modelos e o artefato circular é facilmente visualizado na imagem de di-

ferença de modelos. O *dataset* utilizado possui uma tendência de conter uma falha geológica próxima da região em que o artefato ocorre e a rede generaliza este elemento em todos os modelos preditos.

Em geral as redes apresentam predição satisfatória quando o modelo de velocidades é mais simples (Figura 4a) e com menor falhas geológicas. Modelos de velocidade contendo dobramentos (Figura 4b) também são satisfatoriamente preditos pela RNA, embora a informação da interface de camadas se perca por conta dos gradientes preditos. Modelos em que existem dobramentos somados à uma falha geológica (Figura 4c) apresentam menor acurácia na predição da rede. Nesses casos mais complexos os modelos preditos são capazes de identificar os dobramentos porém as falhas se confundem. Ainda assim é possível visualizar um princípio de identificação dessas características geológicas e o treinamento da rede por mais épocas e com um *dataset* maior, contendo mais amostras com falhas e dobramentos, pode favorecer a identificação destas estruturas.

É possível analisar a similaridade entre as imagens dos modelos preditos pela rede e os modelos reais utilizados como *input* para a geração dos tiros sísmicos. A Figura 5 apresenta o coeficiente de Sorensen-Dice de cada *fold* durante a *cross-validation* da rede, para as duas arquiteturas utilizadas. Nota-se que a similaridade obtida pela rede UNet modificada apresenta maior estabilidade no coeficiente de similaridade entre os *folds*, enquanto que a rede UNet não modificada é menos estável.

A similaridade de aproximadamente 75% evidencia as diferenças entre as imagens provocadas pela presença de gradientes nos modelos preditos, visto que os limites das camadas dos modelos utiliza-

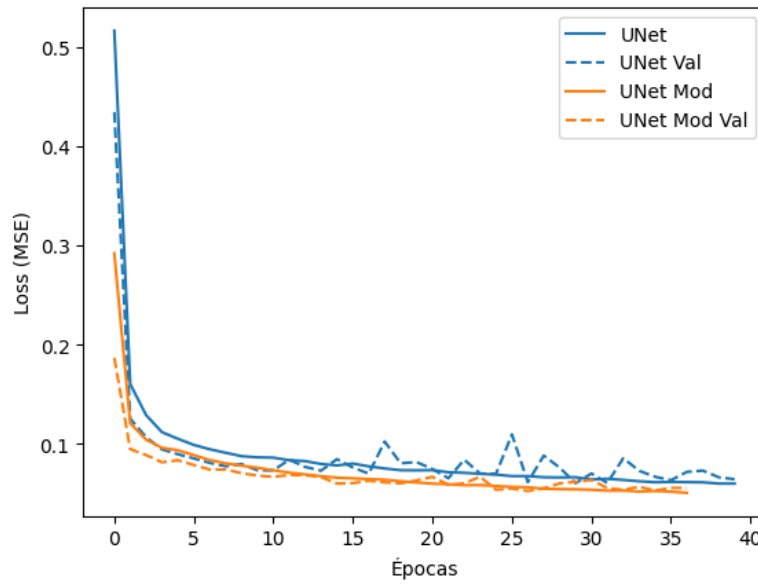
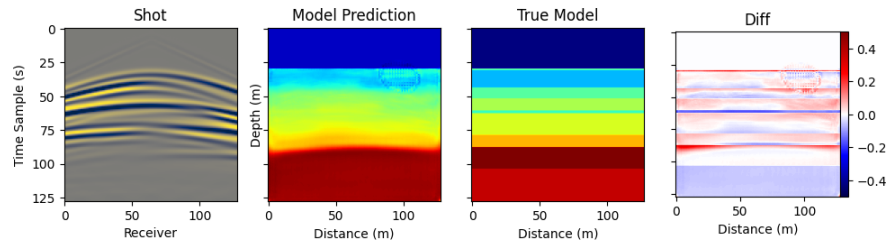
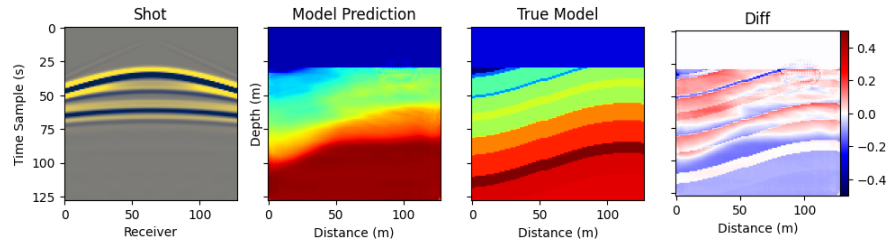


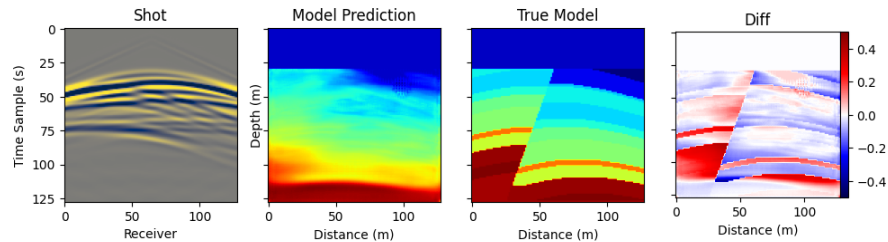
Figura 3: Curvas de *loss* para as duas arquiteturas de redes treinadas.



(a) Amostra simples, sem falhas geológicas.



(b) Amostra com dobramento.



(c) Amostra com falha geológica e dobramento.

Figura 4: Comparação entre os tiros sísmicos de *input* da rede, com os modelos preditos, modelos reais e a diferença (subtração) entre os modelos preditos e reais.



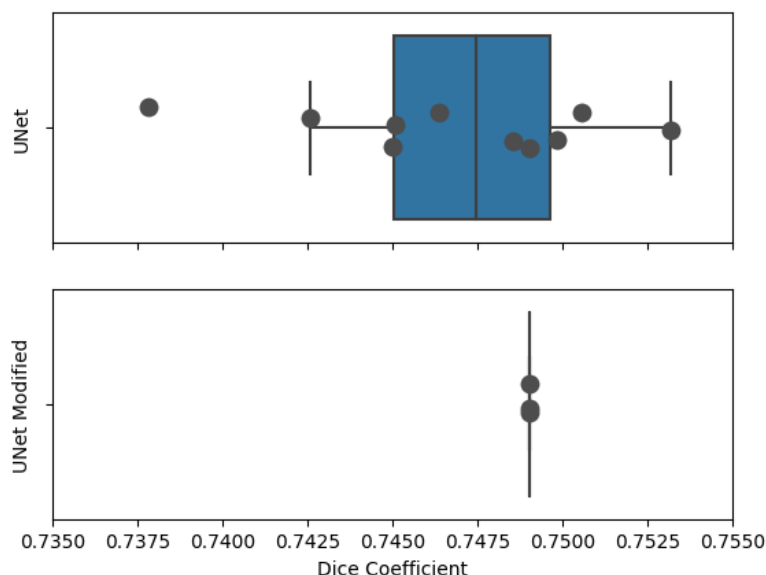


Figura 5: Coeficiente de Sorensen-Dice para as redes treinadas, um indicativo para a estabilidade da similaridade entre modelos reais e preditos pelas redes.

dos como *input* para a simulação dos tiros sísmicos são bem definidos e a interface entre elas é descontínua.

Otimizações das redes podem ser feitas com pequenas alterações das arquiteturas ou da própria geração dos tiros sísmicos a partir do modelo de velocidades, de forma que a predição se mostre mais precisa. O próprio *dataset* utilizado é um fator limitante para o treinamento e validação da rede e a ampliação do número de modelos de *input* devem melhorar a eficiência da rede. Ainda assim, as soluções encontradas pela rede se mostram estáveis.

## 5 Conclusão

As RNAs se mostram como soluções promissoras para o desafio da inversão de modelos de velocidade e mapeamento do assoalho oceânico; onde independente da arquitetura de UNet utilizada, conseguiu-se resultados satisfatórios. Em futuros trabalhos, espera-se investigar os artefa-

tos circulares que aparecem em algumas predições. Outro próximo passo seria introduzir uma etapa *Physics Informed* para otimização da rede para modelos complexos; e paralelamente e gerar mais dados para treinamento, estimulando a capacidade de generalização das RNAs.

## 6 Agradecimentos

Deixamos nossos agradecimentos ao Centro Brasileiro de Pesquisas Físicas pela grandiosa oportunidade de participar da sexta edição da Escola Avançada de Física Experimental do CBPF, uma experiência enriquecedora acadêmica, interpessoal e profissionalmente. Em especial, agradecemos aos professores e monitores do módulo de Inteligência Artificial Aplicada.

## Referências

- [1] KEAREY, P.; BROOKS, M.; HILL, I. **Geofísica de exploração**. Tradução: Maria Cristina Moreira Coelho, São Paulo. Oficina de Texto, p. 438, 2009.
- [2] FLECK, Leandro, et al. **Redes neurais artificiais: Princípios básicos**. Revista Eletrônica Científica Inovação e Tecnologia, 2016, 1.13: 47-57.
- [3] BISHOP, C. M. **Neural networks for Pattern Recognition**. Clarendon Press, Oxford, 1995.
- [4] BRAGA, A. de.; LUDERMIR, T. B.; CARVALHO, A.C. P. de L. F. **Redes Neurais Artificiais teoria e aplicações**, Rio de Janeiro: LTC – Livros Técnicos e Científicos Editora S.A., 2000.
- [5] HAYKIN, S. **Redes Neurais - Princípios e Práticas**. BOOKMAN, São Paulo, 2ª ed. 2001. 900 p.
- [6] CHOLLET F. **Deep learning with Python**. v. 53. 2018.
- [7] RONNEBERGER, O.; FISCHER, P.; BROX, T. **U-net: Convolutional networks for biomedical image segmentation**. In: **Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18**. Springer International Publishing, p, 234-241, 2015.
- [8] DROZDZAL, M. et al. **The importance of skip connections in biomedical image segmentation**. In: **International Workshop on Deep Learning in Medical Image Analysis, International Workshop on Large-Scale Annotation of Biomedical Data and Expert Label Synthesis**. Springer, Cham, p. 179-187, 2016.
- [9] CAMPOS, Luan Rios. **Estimando o modelo inicial de velocidades para fwi usando deep learning**. IV Seminário de Avaliação de Pesquisa Científica e Tecnológica, 2019. n. 86.
- [10] ARAYA-POLO, Mauricio et al. **Deep-learning tomography**. The Leading Edge, v. 37, n. 1, p. 58-66, 2018.
- [11] TARGINO, Jonas Mendonça, et al. **A Deep-Learning inversion method for seismic velocity model building**. In: **First EAGE Conference on Machine Learning in Americas**. EAGE Publications BV, 2020. p. 1-5.
- [12] RICHARDSON, Alan. **Deepwave**. 2022. Disponível em <https://ausargeo.com/deepwave/> acesso em 13 de Marco de 2022
- [13] GERON, A. **Hands-On Machine Learning With Scikit-Learn & Tensor Flow**. Califórnia: O’Riley, Ed. 2, 2019.