

Obs: depois passamos tudo pro tex!!! Só um rascunho pra escrever ideias

Zap da Ana: (21) 98896-2609

<https://research.google.com/colaboratory/local-runtimes.html>

Roteiro

1) O dados sísmico será gerado com o Deep Wave <https://github.com/ar4/deepwave>.

No notebook 01-gera_shots_sample.ipynb temos um exemplo de uso do deep wave onde geramos os tiros e plotamos.

2) O gerador de modelos está no notebook 02-gera_modelos.ipynb. Nele é possível controlar o grau de complexidade dos modelos e das estruturas que estarão presentes. Ele se baseia em premissas geológicas, como deposição, falha, dobramentos e erosão. Experimente.

3) A geração dos tiros a partir dos modelos está sistematizada em 03-gera_shots.ipynb. Para o treinamento da rede fazer sentido, você deve ter cuidado em gerar pares de modelo-dado sísmico com numeração coerente.

4) Rodar o notebook 04a-treino_rede.ipynb. Nele temos os trechos de código que combinados fazem o treinamento da rede neural. Temos 2 opções de arquitetura: unet e unet_mod.

a) Treinar as redes utilizando os Modelos1 e Modelos1a. Qual parece ter o resultado melhor? A que fator você atribui o resultado melhor? Dica, verificar os intervalos de velocidade nos modelos.

b) Treine a rede utilizando a U-Net modificada com o conjunto Modelos1a. Melhorou? Qual diferença você observa entre as duas arquiteturas?

Tente treinar mais de uma vez. O resultado se mantém igual? A quais fatores você atribui a variação entre os resultados de treinamentos independentes?

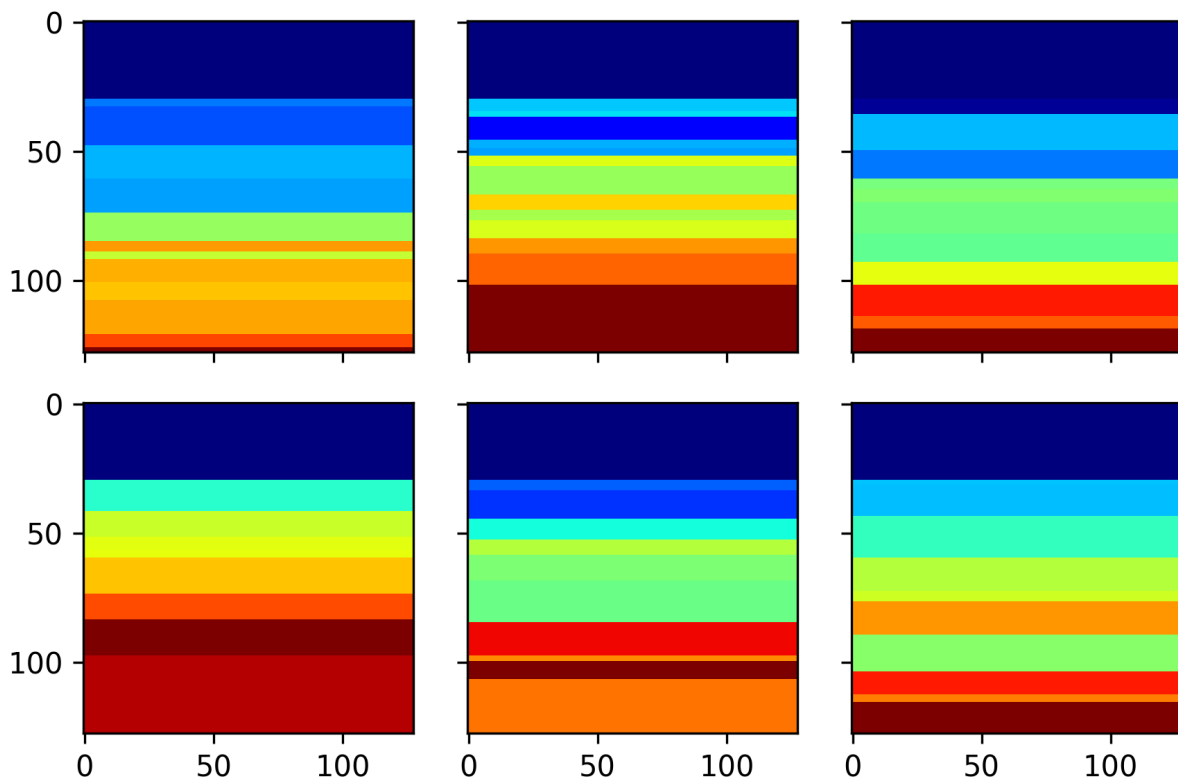
c) Use as redes nas versões com função de ativação sigmoide. Por que podemos usar essa função de ativação no lugar da linear?

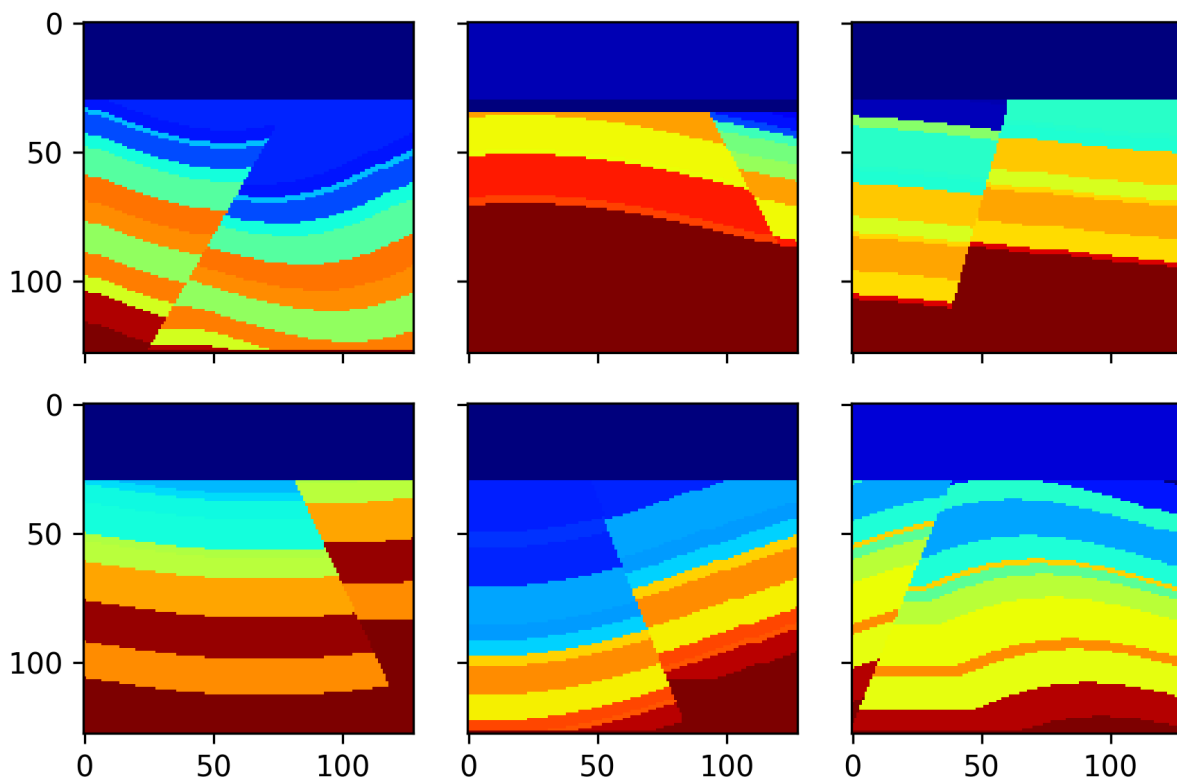
5) Agora tente treinar com modelos que não sejam compostos de camadas horizontais. O que acontece?

Essas perguntas são uma sugestão de roteiro. Fiquem bem à vontade para testar outras arquiteturas, funções loss, etc.

Gerador de Modelos

Os modelos de velocidades são gerados considerando diferentes características geológicas, como falhas, processos de erosão e quebras e as próprias velocidades mínimas e máximas. Dois conjuntos de modelos de velocidades são criados, cada um contendo três grupos. O primeiro conjunto se baseia em modelos simples, sem defeitos geológicos e foi considerado no escopo deste trabalho como o conjunto simples. O segundo conjunto inclui defeitos geológicos e apresentam maior complexidade, sendo considerado o conjunto complexo. Para os dois conjuntos também foram gerados modelos com intervalos de velocidades diferentes.



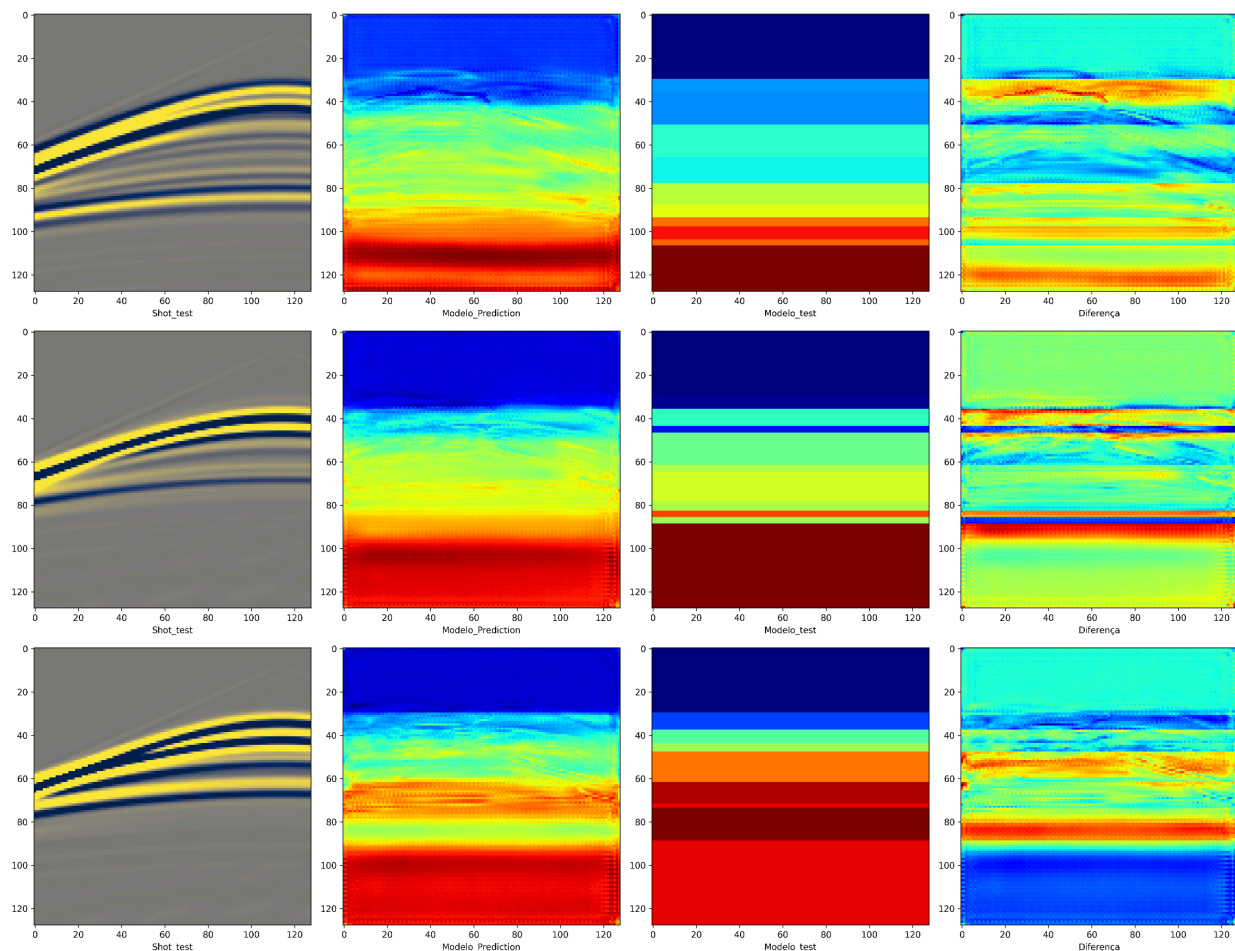


Geração de Tiros

O loop de geração de modelos foi rodado em dois momentos. Os conjuntos gerados serão nomeados como “modelos iniciais” e “modelos secundários”. A partir dos modelos iniciais um conjunto de *shots* foram gerado utilizando o pacote *DeepWave* e, a partir dos modelos secundários, outros dois conjuntos de *shots* foram gerados, sendo o primeiro com parâmetros de simulação idênticos ao gerado para os modelos iniciais e o segundo com uma frequência 25% maior. A Tabela 1 apresenta os parâmetros de cada conjunto de dados gerados.

Parameters	Group 1	Group 2	Group 3
Peak Frequency	8 Hz (?)	8 Hz (?)	10 Hz (?)

Cada conjunto de *shots* é um input para a rede neural. O primeiro conjunto é utilizado para o treinamento da rede. O segundo é utilizado para o teste da rede, com um método de geração de *shots* idêntico, porém com modelos de velocidades diferentes. O terceiro é utilizado para o último teste da rede, com o mesmo modelo de velocidades do segundo conjunto porém *shots* de frequência 25% maior.



Obs.: precisa adicionar uma barra de cores na diferença entre as duas figuras para facilitar a visualização!!!

Desempenho da arquitetura para os conjuntos de modelos

Otimização de Hiperparâmetros

Modificações na arquitetura

A introdução de uma camada extra inicial na rede, para o tratamento preliminar das imagens “quebrou” as métricas propostas (*loss*, *dice*, MSE). A camada extra fazia uma convolução 2D e um MaxPooling sucedido de um Dropout. Não encontramos motivo pra isso 😞

Usamos o Super PC do CBPF e quase maximizamos a memória. Idealmente devemos aumentar o Batch Size!