

# Simulação de Equação de Onda Unidimensional Utilizando Redes Neurais Recorrentes

Lavínia Mitiko Takarabe

*Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas, Universidade Federal de ABC*  
São Bernardo do Campo, Brazil  
ORCID: 0000-0001-8493-7033

Carolina Benetti

*Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas, Universidade Federal de ABC*  
São Bernardo do Campo, Brazil  
ORCID 0000-0003-0304-0473

Mauricio Devinentis

*Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas, Universidade Federal de ABC*  
São Bernardo do Campo, Brazil  
ORCID: 0000-0001-9279-8213

Anderson Gabriel Santiago

*Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas, Universidade Federal de ABC*  
São Bernardo do Campo, Brazil  
ORCID: 0000-0001-7581-1148

**Abstract**—Este artigo apresenta um estudo de Redes Neurais Recorrentes para a solução da Equação de Onda unidimensional considerando a condição de contorno de Neumann. São consideradas duas arquiteturas de Redes Recorrentes: LSTM e GRU. Os dados utilizados nas etapas de treinamento e teste foram calculados utilizando o Método de Diferenças Finitas no Domínio do Tempo e são exploradas diversas combinações de hiperparâmetros, resultando em análises quantitativas e qualitativas, bem como a observação da variação de cada hiperparâmetro individualmente em relação às características de amplitude, fase e frequência simuladas. Como função custo, foi adotado o Erro Quadrático Médio, com otimizador Adam e foram observados erros da ordem de  $10^{-4}$

**Index Terms**—Redes Neurais Recorrentes, GRU, LSTM, Equação de onda e Método de Diferenças Finitas.

## I. INTRODUÇÃO

As técnicas relacionadas ao uso do ultrassom e suas ondas constituem um sistema de monitoramento de saúde que é, na maioria de suas aplicações, considerado não-invasivo e ao mesmo tempo inclusivo, i.e., não há a necessidade de “invadir” o corpo do paciente e não há a imposição de restrições para realização de exames.

No âmbito da Engenharia Biomédica, diversos estudos foram realizados para melhorar as análises desta técnica, dentre os quais se encontram o uso de redes neurais para predição da propagação de uma onda de ultrassom [1] e a análise das Séries Temporais das ondas de ultrassom para identificar alterações em tecidos, como visto em [2]. De modo a evidenciar que simulações computacionais da Equação de Onda são necessárias para fins de comparação e verificação de equipamentos.

Nos últimos anos, os algoritmos de Redes Neurais (“*Neural Networks*” - NN) ganharam uma certa notoriedade devido a sua elevada capacidade de resolver problemas complexos

como, por exemplo, processamento de imagens médicas e reconhecimento de fala e imagens. Além disso, eles se firmaram como uma ferramenta muito poderosa no campo do Cálculo Numérico: uma das primeiras implementações de redes neurais focando neste tipo de problema foi apresentada por Lee e Kang [3] visando a otimização de parâmetros do Método das Diferenças Finitas (“*Finite Difference Method*” - FDM) utilizado para resolver Equações Diferenciais Parciais (“*Partial Differential Equations*” - PDEs) [4].

Desde então, foram propostos novos modelos de arquitetura de redes neurais buscando avaliar equações diferenciais, como as equações de onda [5] e a equação eletrônica de Schrödinger [6]. Como esperado pelo Teorema Universal da Aproximação, que indica que qualquer função contínua pode ser aproximada por uma NN com uma camada oculta (*hidden layer* - HL) e um número finito de neurônios [4], estes estudos mostraram resultados sólidos.

Recentemente, arquiteturas específicas de redes neurais foram desenvolvidas visando solucionar os problemas das Séries Temporais, como a Rede Neural Recorrente (RNN). A RNN é caracterizada pela sua capacidade elevada de processar dados sequenciais, como discutido em Goodfellow [7], além de ser adequada para modelar soluções de equações de onda [5].

Em seu formato original, as redes neurais recorrentes simples (Simple-RNNs) possuem um problema caracterizado pela memória de curto-prazo, o que indica que, a partir de um determinado instante de tempo, a rede irá falhar em sua previsão de novos valores, se estes dependerem de informações de etapas passadas. Duas soluções possíveis são dadas a partir do desenvolvimento de duas arquiteturas de rede, conhecidas como *Gated Recurrent Unit* (GRU) e *Long Short-Term Memory* (LSTM) [8].

Ambas arquiteturas apresentam conceitos similares e visam resolver o problema da memória de curto prazo implementando um mecanismo interno de funções de ativação específicas, conhecido como *gate*, que é responsável por realizar o aprendizado sequencial de uma maneira inteligente: compreendendo qual informação a rede precisa manter ou descartar de acordo com critérios pré-determinados – que podem ser verificados em Goodfellow et al. [7].

A principal diferença entre a GRU e a LSTM se dá pela primeira possuir apenas dois *gates* em sua estrutura, o *Update Gate* e o *Reset Gate*, enquanto a segunda possui dois *gates* adicionais além destes, que seriam o *Forget Gate* e o *Output Gate* [8]. É importante ressaltar que esses *gates* também são redes neurais artificiais com características próprias.

Estas arquiteturas têm sido utilizadas para solucionar sistemas de equações diferenciais utilizando Métodos Numéricos, como o Método de Diferenças Finitas no Domínio de Tempo (*Finite Differences - Time Domain* - FDM-TD) [9]. Partindo deste princípio, este artigo apresenta uma comparação do desempenho entre a RNN-GRU e a RNN- LSTM na previsão da solução de uma equação de onda unidimensional para meios lineares e homogêneos e para as condições de contorno de Neumann, definidas para o gradiente da variável primária. O conjunto de dados utilizado para as etapas de treinamento e validação foi calculado utilizando FDM-TD e os seguintes hiperparâmetros foram considerados: quantidade amostras simuladas (*output dimension* - OD); quantidade de unidades de aprendizado; quantidade de amostras por época de treinamento (*batch*); e função de ativação.

## II. METODOLOGIA

### A. Preparação do Conjunto de Dados e Solução da Equação de Ondas

A implementação do FDM-TD e das arquiteturas da RNN foi realizada em Python 3.8 distribuição Anaconda em conjunto com os pacotes *Numpy*, *Tensorflow* e *Matplotlib*. A equação de onda unidimensional, suas condições de contorno e suas condições iniciais são dadas pelas equações 1 a 5:

$$\frac{\partial^2 u(x, t)}{\partial t} = c^2 \frac{\partial^2 u(x, t)}{\partial x^2} \quad (1)$$

$$u(x = 0, t) = \sin(\omega t) \quad (2)$$

$$\frac{\partial u}{\partial x}(x = L, t) = 0 \quad (3)$$

$$u(x, 0) = 0 \quad (4)$$

$$\frac{\partial u(x, 0)}{\partial t} = 0 \quad (5)$$

Como o principal objetivo deste artigo é comparar as arquiteturas GRU e LSTM na predição da solução de equações de onda, um conjunto de parâmetros de onda padronizado foi considerado: período  $T = 1$  (s), comprimento de onda

$\lambda = 1(m)$  e velocidade de onda plana  $c = 1(m/s)$ ; e para o modelo de FDM-TD,  $\Delta x = \lambda/20(m)$ ,  $\delta t = T/20(s)$ ,  $L = 2\lambda(m)$  com uma janela de tempo de  $2T$  (s).

### B. Etapas de Teste e Validação e Configuração da Rede Neural

A solução do FDM-TD resultou em uma matriz  $U$  de dimensão  $(N,M)$  que foi transformada em um vetor  $(NM, 1)$ , como ilustrado na Figura 1.

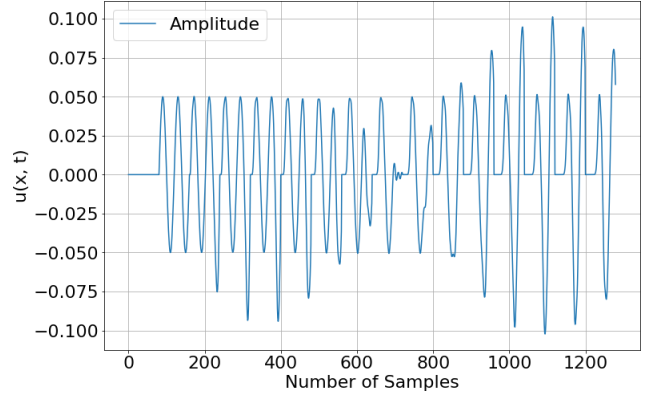


Fig. 1. Solução de equação de onda obtida pelo método de diferenças finitas utilizada como conjunto de dados para treinamento e validação.

Para as etapas de Treino e Validação, uma proporção de 95:5 foi utilizada, isto é, 95% dos dados disponíveis para treinamento (1292 amostras) e 5% para validação (68 amostras). Visando uma melhora nos resultados de treino e da validação, foi considerada uma etapa de pré-processamento, que consistiu em uma normalização por min-max. Após a conclusão dos procedimentos de treino e validação, foi aplicada a desnormalização do conjunto de dados para obtenção dos valores reais previstos.

A função custo considerada neste artigo foi o erro quadrático médio (*Mean Squared Error* - MSE), e o otimizador adotado foi o de Adam [10] com uma taxa de aprendizado adaptativa. Para cada arquitetura, o treinamento teve uma série de parâmetros ajustáveis como apresentado na Tabela II-B juntamente com seus intervalos de valores (ou valores possíveis). É importante ressaltar que o mesmo conjunto de hiperparâmetros foi considerado para ambas arquiteturas, de modo a permitir a comparação de seus desempenhos.

TABLE I  
CONJUNTO DE HIPERPARÂMETROS E INTERVALO DE VALORES

Hiperparâmetros	Intervalo de valores
Nt	[2, 10]
Nu	[100, 500]
Length	[1, 50]
Batch	1, 5, 10, 25, 50

As funções de ativação consideradas são ilustradas na Figura 2 a seguir:

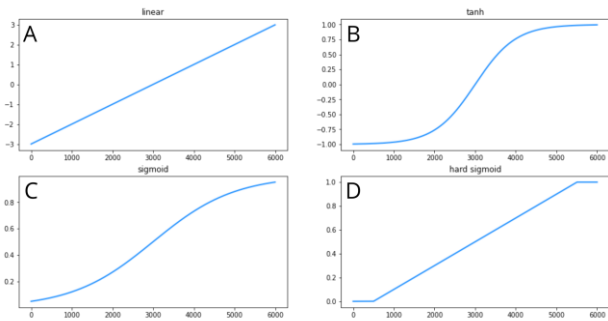


Fig. 2. Funções de Ativação consideradas: A) Linear; B) Tanh; C) Sigmoid; D) Hard\_Sigmoid.

Para a otimização dos hiperparâmetros, foi considerada uma exploração inicial aleatória, levando em conta a função custo avaliada para os conjuntos de dados de treinamento e validação. Após a definição de um intervalo subótimo de cada hiperparâmetro, foi implementada a técnica de *grid search*<sup>1</sup>, em que os hiperparâmetros variaram de forma progressiva e igualmente espaçada. É importante notar que o número de épocas utilizado nos experimentos não foi considerado como um hiperparâmetro, pois foi possível monitorar o erro de validação utilizando método de *early\_stopping*<sup>2</sup>.

### III. RESULTADOS

Essa seção apresenta os resultados considerando a funções de ativação tanh que apresentou os menores valores de erro para os conjuntos de dados de treino e validação.

Ambas arquiteturas apresentaram valores comparáveis do evaluation time das etapas de treino e validação, que variou entre 10-80(s/época) com um tempo médio de 52(s/época). Além disso, o mesmo conjunto de hiperparâmetros foi aplicado para GRU e a LSTM, estes podem ser verificados na Tabela III.

TABLE II  
CONJUNTO ÓTIMO DE HIPERPARÂMETROS

Parâmetros	<i>tanh</i>
Nt	8
Nu	100
Batch	1
Length	10

#### A. LSTM

A Tabela 3 apresenta os valores da função custo para os conjuntos de dados de teste e validação e a Fig. 3 ilustra uma comparação qualitativa entre a solução por FDM-TD e a predição da RNN-LSTM para a função de ativação e o conjunto de hiperparâmetros que apresentaram o menor valor de erro para a função custo (MSE).

As Figuras 4, 5, 6 e 7 ilustram a variação da onda predita pelo modelo de rede neural conforme os hiperparâmetros Nt, Nu, Length e batch, respectivamente, sofreram variações.

<sup>1</sup>[https://scikit-learn.org/stable/modules/grid\\_search.html](https://scikit-learn.org/stable/modules/grid_search.html)

<sup>2</sup>[https://keras.io/api/callbacks/early\\_stopping/](https://keras.io/api/callbacks/early_stopping/)

TABLE III  
ERRO MÍNIMO DA FUNÇÃO CUSTO PARA A LSTM

Dataset	<i>tanh</i>
Treino	33E-6
Validação	44E-6

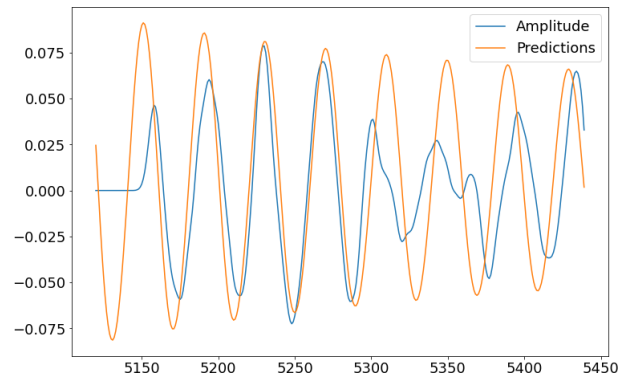


Fig. 3. Desempenho da LSTM (em laranja) na predição da solução da equação de onda (em azul) para função de ativação *tanh*.

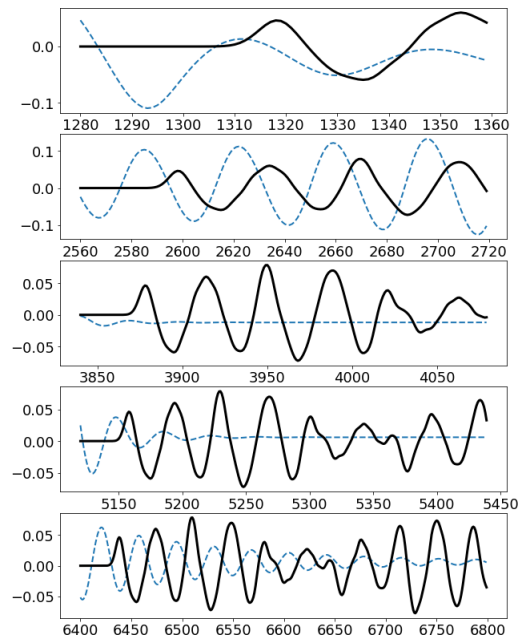


Fig. 4. Comparação das predições geradas pela RNN-LSTM (em azul) mediante variação do hiperparâmetro Nt em um intervalo de [2,10] com passo 2, para a solução da equação de onda simulada pelo FDM-TD (em preto).

#### B. GRU

A Tabela III-B apresenta os valores da função custo para os conjuntos de dados de teste e validação e a Fig. 8 ilustra uma comparação qualitativa entre a solução por FDM-TD e a predição da RNN-GRU para a função de ativação e o conjunto de hiperparâmetros que apresentaram o menor valor de erro para a função custo (MSE).

As Figuras 9, 10, 11 e 12 ilustram a variação da onda predita pelo modelo de rede neural conforme os hiperparâmetros Nt,

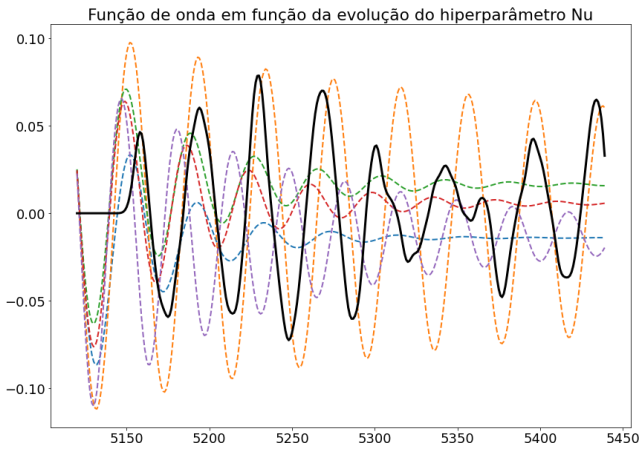


Fig. 5. Comparação das previsões geradas (curvas tracejadas) pela RNN-LSTM mediante variação do hiperparâmetro  $Nu$  – em um intervalo de  $[100, 500]$  com um passo 100 – para a solução da equação de onda simulada pelo FDM-TD (em preto). As curvas tracejadas representam os seguintes valores de  $Nu$ : Azul – 100, Laranja – 200, Verde – 300, Vermelho – 400 e Roxo – 500.

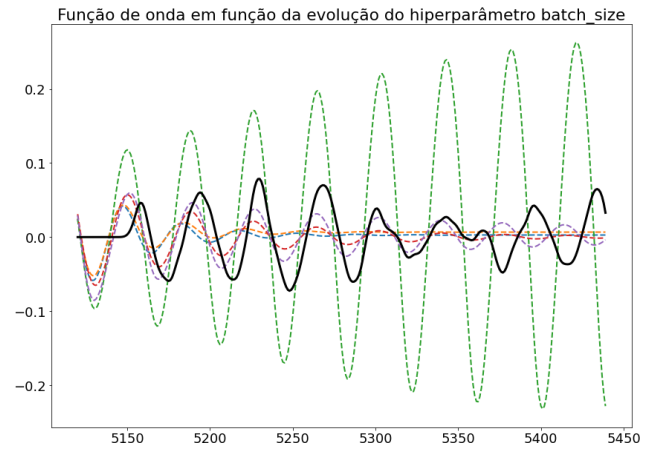


Fig. 7. Comparação das previsões geradas (curvas tracejadas) pela RNN-LSTM mediante variação do hiperparâmetro  $batch$  para a solução da equação de onda simulada pelo FDM-TD (em preto). As curvas tracejadas representam os seguintes valores de  $batch$ : Azul – 1, Laranja – 5, Verde – 10, Vermelho – 25 e Roxo – 50.

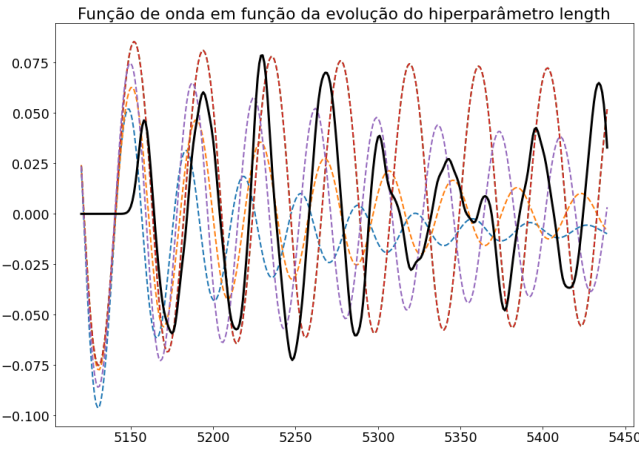


Fig. 6. Comparação das previsões geradas (curvas tracejadas) pela RNN-LSTM mediante variação do hiperparâmetro  $length$  – em um intervalo de  $[10, 50]$  com um passo 10 – para a solução da equação de onda simulada pelo FDM-TD (em preto). As curvas tracejadas representam os seguintes valores de  $length$ : Azul – 10, Laranja – 20, Verde – 30, Vermelho – 40 e Roxo – 50.

$Nu$ ,  $Length$  e  $batch$ , respectivamente, sofreram variações.

#### IV. DISCUSSÃO

Observando as Figuras 4 a 6 e as Figuras 9 a 11, pode-se concluir que os hiperparâmetros  $Nu$ ,  $Nt$  e  $Length$  impactam nas seguintes características da onda simulada: para  $Nt$  observa-se uma variação na amplitude da onda: seu aumento leva a diminuição drástica da amplitude de onda resultante; já para  $Nu$  e  $Length$ , observa-se variações nas amplitudes das ondas porém frequências constantes próximas do ponto de vista qualitativo, da onda simulada. Este comportamento pode ser observado em ambas as arquiteturas consideradas.

Já para o hiperparâmetro  $Batch$  (Figs. 7 e 12), resultou em diferentes comportamentos na predição das ondas. É possível observar que para LSTM, sua variação foi de grande impacto

TABLE IV  
ERRO MÍNIMO DA FUNÇÃO CUSTO PARA A GRU

Dataset	$tanh$
Treino	113E-6
Validação	920E-6

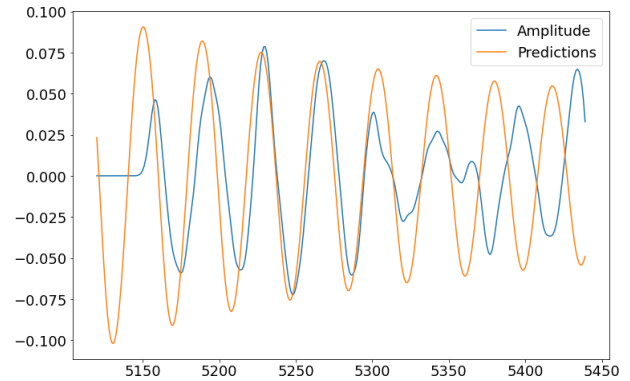


Fig. 8. Desempenho da GRU (em laranja) na predição da solução da equação de onda (em azul) para função de ativação  $tanh$ .

na amplitude das ondas, enquanto na GRU, apesar de também apresentar um impacto significativo nas amplitudes das ondas, este hiperparâmetro também mostrou estar associado a uma variação de frequência.

Ainda considerando o  $Batch$ , foi possível observar graficamente uma maior proximidade do resultado via RNN- GRU com os valores calculados via FDM-TD. Entretanto, o melhor resultado obtido em termos qualitativos (gráficos) e quantitativos (erro da função custo) foi uma rede com arquitetura RNN-LSTM e hiperparâmetros  $Nt = 8$ ,  $Nu = 100$ ,  $batch = 1$  e  $Length = 30$ .

Além disso, é interessante evidenciar que para a LSTM a variação do hiperparâmetro  $Length$  se mostrou ser a mais

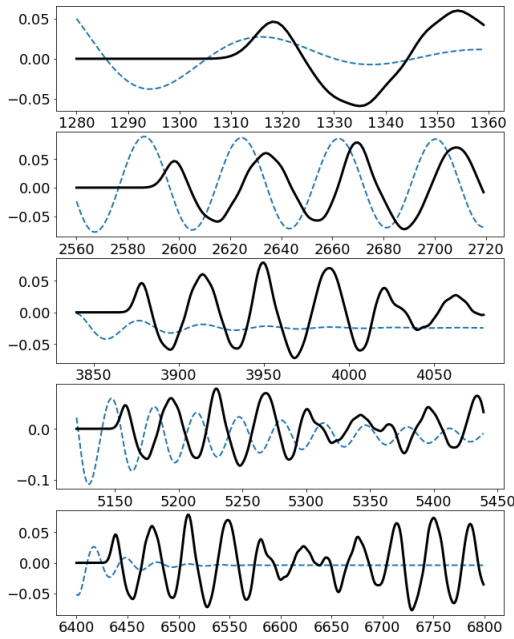


Fig. 9. Comparação das previsões geradas pela RNN-GRU (em azul) mediante variação do hiperparâmetro  $N_t$  em um intervalo de  $[2, 10]$  com passo 2, para a solução da equação de onda simulada pelo FDM-TD (em preto).

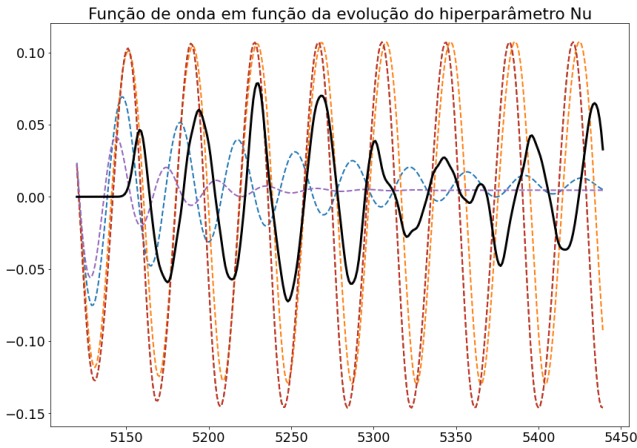


Fig. 10. Comparação das previsões geradas (curvas tracejadas) pela RNN-GRU mediante variação do hiperparâmetro  $Nu$  – em um intervalo de  $[100, 500]$  com um passo 100 – para a solução da equação de onda simulada pelo FDM-TD (em preto). As curvas tracejadas representam os seguintes valores de  $Nu$ : Azul – 100, Laranja – 200, Verde – 300, Vermelho – 400 e Roxo – 500.

importante para ajustar a curva a solução, enquanto para a GRU a variação do hiperparâmetro  $batch$  demonstrou efeitos mais significativos de ajuste.

## V. CONCLUSÃO

Este artigo apresentou um estudo do desempenho de redes neurais recorrentes LSTM e GRU na previsão de valores para a equação de onda unidimensional com condição de contorno de Neumann com uma entrada de onda senoidal. Os resultados apresentados sugerem que ambas arquiteturas podem ser

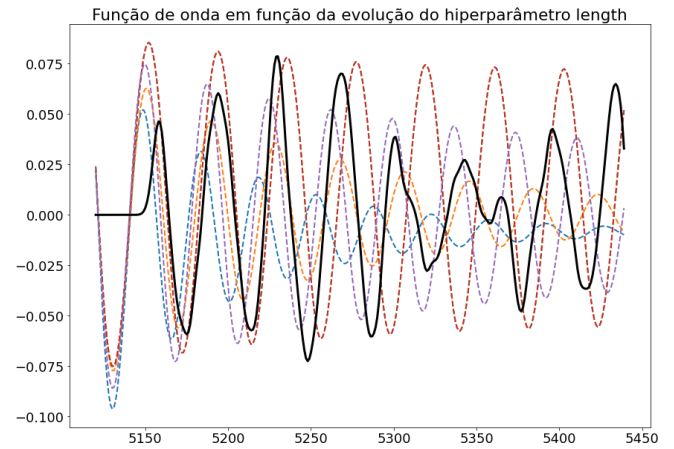


Fig. 11. Comparação das previsões geradas (curvas tracejadas) pela RNN-GRU mediante variação do hiperparâmetro  $length$  – em um intervalo de  $[10, 50]$  com um passo 10 – para a solução da equação de onda simulada pelo FDM-TD (em preto). As curvas tracejadas representam os seguintes valores de  $length$ : Azul – 10, Laranja – 20, Verde – 30, Vermelho – 40 e Roxo – 50.

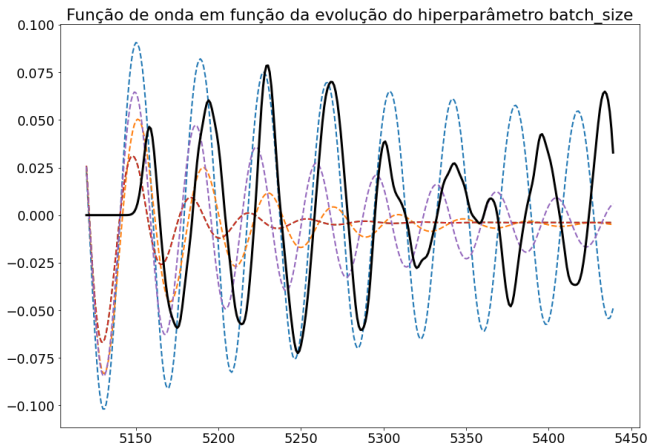


Fig. 12. Comparação das previsões geradas (curvas tracejadas) pela RNN-GRU mediante variação do hiperparâmetro  $batch$  para a solução da equação de onda simulada pelo FDM-TD (em preto). As curvas tracejadas representam os seguintes valores de  $batch$ : Azul – 1, Laranja – 5, Verde – 10, Vermelho – 25 e Roxo – 50.

utilizadas para simular equações de onda, embora ainda sejam necessários ajustes associados a escolha de uma função custo mais adequada para que seja possível encontrar parâmetros que possam ajustar os resultados com maior precisão.

Além disso, identificamos que para ajustar a solução da RNN-LSTM à obtida por FDM-TD deve-se focar no ajuste do  $Length$  para uma arquitetura RNN-LSTM e do  $Batch$  para uma arquitetura RNN-GRU. Contudo, mais estudos sobre o impacto da variação dos valores dos hiperparâmetros sobre a amplitude, frequência e fase são necessários para explicar como as redes inclinam às séries temporais consideradas.

A partir da otimização dos modelos de RNN para a previsão de equações de onda, uma miríade de aplicações torna-se possíveis, como, por exemplo, a implementação destes métodos para refinar a previsão da propagação de uma onda

de ultrassom e/ou identificação de alterações em tecidos.

#### REFERENCES

- [1] Subhayan De, Bhuiyan Shameem Mahmood Ebna Hai, Alireza Doostan, Markus Bause. Prediction of ultrasonic guided wave propagation in solid-fluid and their interface under uncertainty using machine learning. 2021. fihal-03207450f
- [2] Uniyal N, Eskandari H, Abolmaesumi P, Sojoudi S, Gordon P, Warren L, Rohling RN, Salcudean SE, Moradi M. Ultrasound RF time series for classification of breast lesions. *IEEE Trans Med Imaging*. 2015 Feb;34(2):652-61. doi: 10.1109/TMI.2014.2365030. Epub 2014 Oct 24. PMID: 25350925.
- [3] H. Lee and I. S. Kang. Neural algorithm for solving differential equations. *Journal of Computational Physics*, vol. 91, n. 1, pp. 110–131, 1990.
- [4] L. Lima. Numerical Solution of PDE's Using Deep Learning. PhD thesis, 2019.
- [5] T. W. Hughes, I. A. Williamson, M. Minkov, and S. Fan. Wave physics as an analog recurrent neural network.
- [6] J. Hermann, Z. Schatzle, and F. No e. Deep-neural-network solution of the electronic schr odinger equation. *Nature Chemistry*, vol. 12, n. 10, pp. 891–897, 2020.
- [7] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.
- [8] M. Nguyen. Illustrated guide to LSTM's and GRU's: A step by step explanation. "<https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21>". [Online; accessed 19-July-2021], 2018.
- [9] H. M. Yao and L. J. Jiang. Machine learning based neural network solving methods for the fdtd method. In *2018 IEEE International Symposium on Antennas and Propagation USNC/URSI National Radio Science Meeting*, pp. 2321–2322. IEEE, 2018.
- [10] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017