INTERFACE TECNOLÓGICA

MÉTODO DE EULER E REDES NEURAIS PARA APROXIMAÇÃO NUMÉRICA DE EQUAÇÕES DIFERENCIAIS ORDINÁRIAS

EULER'S METHOD AND NEURAL NETWORKS FOR THE NUMERICAL APPROXIMATION OF ORDINARY DIFFERENTIAL EQUATIONS

Clayton Suguio Hida –clayton.hida@fatec.sp.gov.br
Faculdade de Tecnologia de Praia Grande (Fatec) – Praia Grande – SP – Brasil

Gilberto Sussumu Hida – gilberto.hida@fatec.sp.gov.br
Faculdade de Tecnologia de São Caetano do Sul (Fatec) – São Caetano do Sul – SP – Brasil

DOI: 10.31510/infa.v20i2.1726
Data de submissão: 06/09/2023
Data do aceite: 16/11/2023
Data da publicação: 20/12/2023

RESUMO

Neste trabalho, propomos o uso de métodos numéricos para resolução de equações diferenciais em conjunto com o uso de técnicas de aprendizado de máquina para a obtenção de soluções aproximadas de equações diferenciais. Mais especificamente, o presente trabalho visa a implementação do Método de Euler em conjunto com modelos de redes neurais para a aproximação de soluções de equações diferenciais ordinárias. Como metodologia, aplicamos o método de Euler para obter valores aproximados da função solução da equação diferencial na proximidade do ponto inicial. Depois, usamos esses pontos para treinar um modelo de rede neural como uma aproximação da solução real em um intervalo estendido. O modelo final apresenta resultas próximos de uma rede neural treinada com pontos da solução exata da equação diferencial e apresenta melhores resultados quando comparados a aplicação direta do método de Euler em todo o intervalo em consideração. Em particular, a análise mostra que o modelo proposto é indicado nos casos em que o método de Euler não possui estabilidade.

Palavras-chave: Aprendizado de máquina. Estabilidade. Problema de valor inicial. Aproximações.

ABSTRACT

In this work, we propose the use of numerical methods to solve differential equations and the use of machine learning techniques to obtain approximate solutions of differential equations. More specifically, the present work aims at the implementation of Euler's Method together with neural network models for the approximation of solutions of ordinary differential equations. As a methodology, we apply Euler's method to obtain approximate values of the solution of the differential equation near the initial point. We then use these points to train a neural network model as an approximation of the real solution over an extended interval. The final model is close to the neural network trained with points of the exact solution of the differential equation and shows to be better when compared to the direct application of the Euler method in the entire interval under consideration. In particular, the analysis shows that the proposed model is indicated in cases where Euler's method lacks stability.

Keywords: Machine learning. Stability. Initial value problem. Approximations.

1 INTRODUÇÃO

Muitos problemas práticos e da natureza podem ser descritos como uma relação entre uma variável e sua variação em relação a outras variáveis. Como exemplos recentes, temos a variação da população contaminada pelo Covid-19 em relação a situação atual de infectados (PATRÃO e REIS, 2022) e estudos relacionados a crescimento de fungos. Nesses casos, temos a presença de um modelo descrito por uma equação diferencial ordinária. Em muitas situações, não existem métodos diretos que nos forneçam a solução de forma explicita da solução. Para esses casos, podemos recorrer a métodos numéricos de equações diferenciais. Tais métodos, fornecem funções que se aproximam da solução real, fornecendo assim bons resultados dentro de um limite de erro estabelecido a priori. Por outro lado, estamos presenciando o avanço de técnicas de inteligência artificial em vários campos de pesquisa. De aplicações práticas como os carros autônomos (BATHLA *et al.,* 2022) e em saúde (LEE e YOON, 2021), como em aplicações dentro da pesquisa acadêmica (CHOI *et al.,* 2020).

Neste trabalho, propomos a aplicação de inteligência artificial, mais especificamente, redes neurais em conjunto com métodos numéricos para resolução de equações diferenciais.

O método numérico para a solução de equações diferenciais que vamos utilizar é o método de Euler. Trata-se de um método de fácil compreensão e de fácil implementação. O principal parâmetro no método é o passo de iteração, que vamos denotar pela letra h. Esse passo de iteração é de essencial importância, pois à medida que esse passo se torna pequeno, as soluções aproximadas dadas por esse método se tornam cada vez melhores localmente. Queremos encontrar uma alternativa justamente quando o passo h não pode ser tomado tão pequeno quanto se queira. Essa limitação pode ocorrer como uma limitação computacional e se reflete basicamente na quantidade de dados que podemos considerar no problema.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção, abordaremos alguns aspectos essenciais de Equações diferenciais ordinárias, métodos numéricos para equações diferenciais e redes neurais que serão essenciais para o desenvolvimento do restante do artigo.

2.1 Equações Diferenciais Ordinárias e Problemas de valores iniciais

Neste trabalho, estamos interessados em problemas de valores iniciais (PVI) da forma $y' = f(x,y),\ y(x_0) = y_0$.

Sobre certas condições, temos teoremas que asseguram a existência e unicidade das soluções (VIANA e ESPINAR, 2021). Esses teoremas, no entanto, não nos fornecem um meio para encontrar as soluções de forma analítica, ou seja, exata. Dessa forma, somos levados a analisar o problema essencialmente em duas maneiras: A primeira é abordar o problema com uma análise qualitativa das equações diferenciais ordinárias (CRONIN, 2007). A segunda opção é utilizar de meios numéricos para a obtenção de aproximações da solução real. Em muitas situações práticas, uma solução aproximada é suficiente para resolver os problemas, dentro de alguma margem de erro pré-estabelecida. Focaremos aqui na segunda opção, ou seja, em obtenção de soluções aproximadas por métodos numéricos. Passaremos agora a descrever o método numérico que vamos utilizar.

2.2 Método de Euler

Um dos primeiros métodos para a resolução numérica de PVI's é o método de Euler. É um método numérico de primeira ordem (BUTCHER, 2016) e baseado na expansão de Taylor da função em torno do ponto inicial. Dado um PVI $y' = f(x,y),\ y(x_0) = y_0$, obtemos uma sequência de pontos iniciando com x_0 e construído indutivamente por $x_{n+1} = x_n + h$, onde h é denominado de passo. A sequência de aproximações obtida pelo método de Euler é dada pela expressão $y_{n+1} = y_n + hf(x_n,\ y_n)$, onde y_n denota o n-ésimo termo da sequência e h é o passo da iteração. O erro da aproximação no método de Euler depende fundamentalmente do comportamento da função $f(x,y)$ nas proximidades do ponto (x_0, y_0) e também do tamanho do passo h (BUTCHER, 2016).

2.3 Redes Neurais e o teorema de aproximação universal

Segundo (ABU-MOSTAFA, 2012), as redes neurais são uma classe de modelos com inspiração no modelo biológico de um neurônio e que tem um considerável sucesso em aplicações. Em nosso estudo, vamos utilizar um modelo de rede neural com uma camada oculta. Essa escolha se deve principalmente pelo importante teorema na teoria de Redes neurais, denominado de Teorema de Aproximação Universal, devido a G. Cybenko, que diz que toda função contínua em um intervalo fechado, pode ser aproximada por uma rede neural com uma

camada oculta. Recomendamos (CYBENKO, 1989) para a definição formal e a prova do teorema.

3 PROCEDIMENTOS METODOLÓGICOS

Neste artigo, propomos o uso de redes neurais para aproximação de soluções de um problema de valor inicial para valores além do ponto inicial. Vamos usar a linguagem Python na IDE Jupyter Notebook.

Considere o seguinte PVI: $y' = f(x, y)$, $y(0) = x_0$. Queremos obter uma aproximação da solução real $y(t)$ no intervalo [0,10]. Vamos construir um modelo, chamado de Euler + Neural da seguinte forma: Através do método de Euler, obtemos uma sequência de pontos $(x_n, \overline{y_n})_n$ que aproximam a solução no intervalo $[0,2]$. Com os dados obtidos, treinamos uma rede neural, como uma aproximação da solução do PVI no intervalo estendido $[2,10]$. Essa rede neural final é o nosso modelo.

Os resultados obtidos são comparados com a solução analítica do PVI. Também usamos como comparação um modelo, chamado de Neural, que é uma rede neural treinada com os valores da solução analítica no intervalo [0,2] e também comparamos com um modelo, chamado aqui de Euler-All, que é obtida pelo método de Euler no intervalo inteiro [0,10]. Abaixo descrevemos de forma mais detalhada as principais etapas da simulação.

3.1 Aplicação do Método de Euler para a obtenção dos pontos de treino.

Em nossa simulação, utilizamos como exemplo a família de equações diferenciais da forma $y' = -ky$, $y(0) = 1$, onde k é um inteiro positivo. A solução analítica desse PVI é dada por $y(x) = e^{-kx}$. Para a obtenção dos pontos pelo Método de Euler, utilizamos a implementação mostrada na Figura 1.

Figura 1 - Implementação do Método de Euler

```
def get_euler_points(funcao, a,b,y0,n_points):
    h = (b-a)/n_points
    points = np.linspace(a,b, n_points)
    y_values = []
    y_values.append(y0)
    y_temp = y0
    for i in np.arange(1, n_points):
        yn = y_temp + h*funcao(points[i], y_temp)
        y_values.append(yn)
        y_temp = yn
    return points, y_values
```

Fonte: autores (2023)

A escolha pelo Método de Euler se deve principalmente pela sua fácil interpretação e implementação. A Figura 2 mostra os 10 primeiros pontos obtidos pelo método de Euler para o caso k=-21.

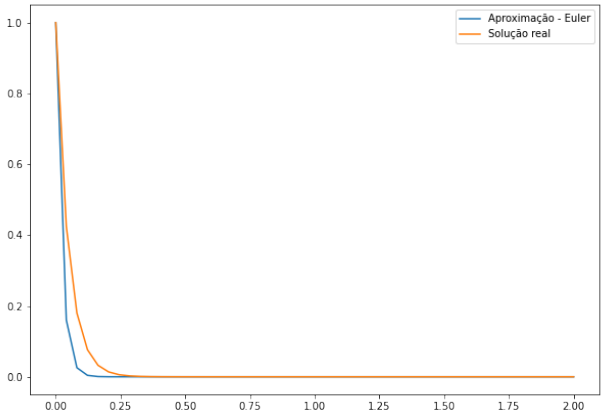
Figura 2 - Resultado da simulação

	xn	yn
0	0.000000	1.000000e+00
1	0.040816	1.600000e-01
2	0.081633	2.560000e-02
3	0.122449	4.096000e-03
4	0.163265	6.553600e-04
5	0.204082	1.048576e-04
6	0.244898	1.677722e-05
7	0.285714	2.684355e-06
8	0.326531	4.294967e-07
9	0.367347	6.871948e-08
10	0.408163	1.099512e-08

Fonte: autores (2023)

Na Figura 3, plotamos a solução obtida pelo método de Euler e os valores reais das funções.

Figura 3 - Comparação Predição e Real



Fonte: autores (2023)

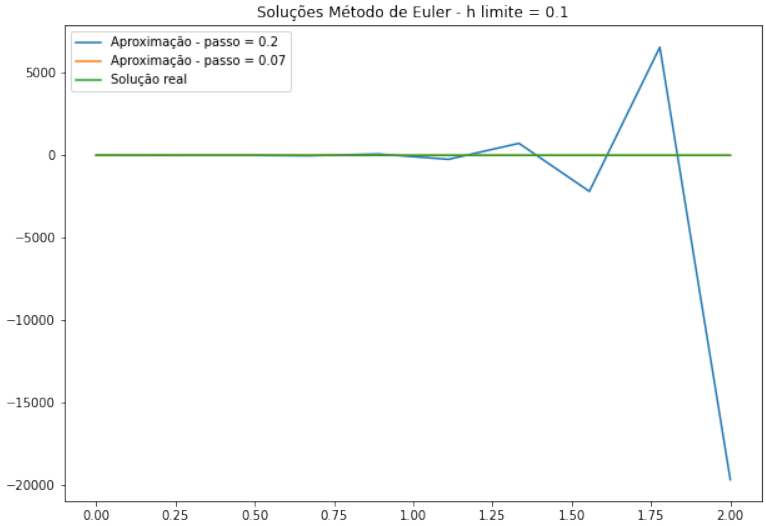
3.2 Escolha da quantidade de pontos e estabilidade

O método de Euler fornece boas aproximações locais para a solução do PVI à medida que o passo h se torna cada vez menor (BURDEN, 2015). Em análise numérica, dizemos que o método de Euler possui estabilidade parcial, pois a convergência da aproximação obtida pelo método para a solução analítica, quando $x \to \infty$ depende fundamentalmente do passo h .

No caso particular das equações diferenciais da forma $y' = -k\,y$, temos estabilidade no caso em que o valor de h fica dentro da região $\left(0,\ \frac{2}{k}\right)$ (BURDEN, 2015).

Como estamos trabalhando em um intervalo fixo, o passo h está intimamente relacionado com a quantidade de pontos. Mais especificamente, a relação é dada por $h = \frac{(b-a)}{n}$, onde a e b são os extremos do intervalo e n é o número de pontos. A Figura 4 mostra a simulação para a equação $y' = -20\,y$ para os casos em que $h = 0{,}2$ e $h = 0{,}07$. Observe que neste caso, o valor limite que assegura a estabilidade da solução é $h = 0{,}1$.

Figura 4 -Método de Euler com diferentes passos



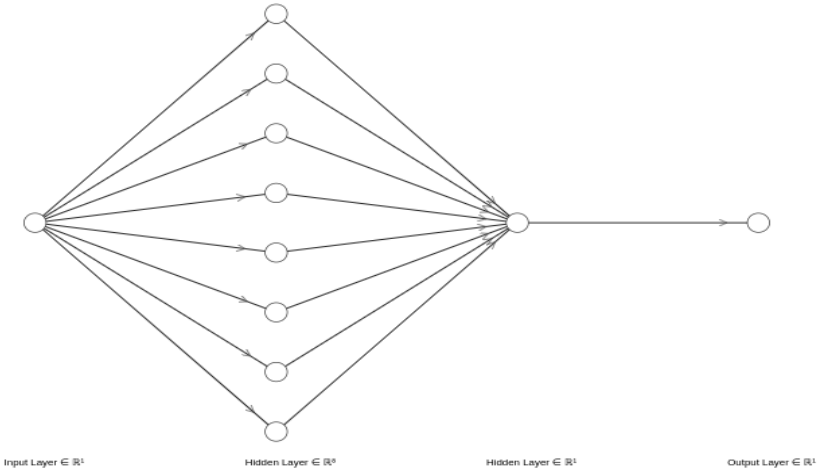
Fonte: autores (2023)

Gostaríamos de observar se nossa abordagem fornece bons resultados para escolhas de passos h onde o método de Euler não fornece bons resultados, ou seja, fora do intervalo $\left(0,\ \frac{2}{k}\right)$. Assim, dada o intervalo $[a, b]$, vamos escolher o número de pontos n de tal forma que $\frac{(b-a)}{n} > \frac{2}{k}$, ou seja, $n < \frac{k(b-a)}{2}$.

3.3 Arquitetura da rede neural

Como observado na seção 2.3, o Teorema de aproximação universal diz que podemos aproximar qualquer função contínua no intervalo $I = [0,1]$, por uma rede neural com uma camada interna. Com base nesse resultado, utilizamos como padrão a arquitetura de rede mostrada na Figura 5. A Figura 6 mostra a arquitetura da implementação da classe.

Figura 5 - Arquitetura de uma rede neural



Fonte: autores (2023)

Figura 6 - Estrutura da classe Neural_Network

```
# Rede Neural
class Neural_Network:
    def __init__(self, nodes, num_epochs, learning_rate):
        self.nodes = nodes
        self.num_epochs = num_epochs
        self.learning_rate = learning_rate

    def sigmoid(self, x):
        return 1 / (1 + np.exp(-x))

    def sigmoid_derivative(self,x):
        return self.sigmoid(x) * (1 - self.sigmoid(x))

    def fit_nn(self, X, Y):
        pass
    # Predição
    def predict(self, X):
        pass
```

Fonte: autores (2023)

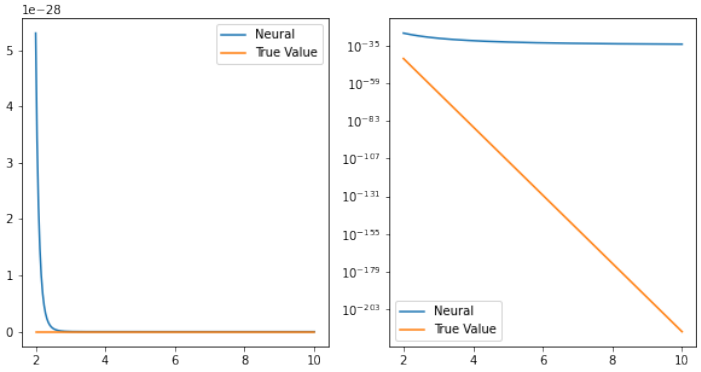
Como parâmetros da classe, temos o número de nodes, o número de *epochs* e o *learning_rate* utilizados para o treino da rede neural. O método fit_nn() é usado para treinar a rede, usando a otimização de gradiente descendente e o método predict() é usado para fazer a predição da rede neural treinada.

4 RESULTADOS E DISCUSSÃO

Consideramos o PVI $y' = -50\,y,\ y(0) = 1$. A Figura 7 mostra os valores preditos pela Modelo Neural no intervalo [2,10], onde a rede foi treinada com 200 pontos obtidos

diretamente da solução analítica no intervalo [0,2]. O gráfico a esquerda mostra o valor real da função (em laranja) e os valores preditos pela rede neural (em azul). O gráfico a direta mostra os mesmos dados, mas em escala logarítmica.

Figura 7 – Modelo Neural e valores reais

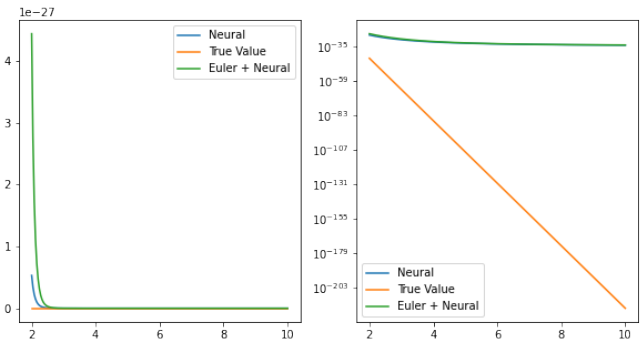


Fonte: autores (2023)

No intervalo de teste [2,10], o erro médio quadrático da aproximação pela rede neural é de $2{,}5 \times 10^{-57}$. A rede neural obtida será usada apenas para efeito de comparação com nosso modelo final, uma vez que para treiná-la, foi necessário o uso de dados da função solução, o que em muitas situações não possuímos. Para tanto, ao invés de usar os dados reais da função no intervalo [0,2], vamos usar os dados obtidos pelo método de Euler.

Obtemos assim o modelo Euler+Neural, com a predição de vaores mostrados na Figura 8.

Figura 8 - Modelo Euler+Neural e valores reais

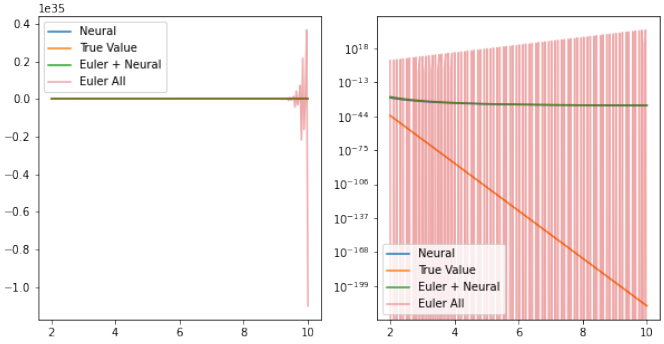


Fonte: autores (2023)

INTERFACE TECNOLÓGICA

Podemos observar que os valores preditos pela Modelo Neural e os valores preditos pelo modelo Euler + Neural são similares no comportamento. O erro médio quadrático da aproximação pela Euler + Neural é de $1{,}6 \times 10^{-55}$ e a diferença entre o modelo Neural e Euler + Neural é de $1{,}2 \times 10^{-55}$. Na Figura 9 completamos com mais um modelo, que é a simples implementação do método de Euler no intervalo [2,10].

Figura 9 - Instabilidade do método de Euler



Fonte: autores (2023)

O comportamento errático apresentado pelo modelo Euler All se deve ao fato que estamos usando como passo o valor $h = 0{,}05$ que está acima do valor limite de estabilidade $h = 0{,}04$ para o exemplo em questão. A Figura 10 mostra o resultado de outras simulações onde variamos o parâmetro k do PVI. Observamos que em todas as situações, o erro quadrático médio do modelo Euler+Neural é menor que o erro quadrático médio do modelo Euler All. Observamos ainda que o maior erro do modelo Euler All ocorre quando o valor do passo h é significante maior que o valor h_lim, que representa o maior valor que h pode tomar para garantir a estabilidade do método de Euler.

Figura 10 - Simulação com outros valores de k e instabilidade

	k	h_lim	h	Erro_Euler+Neural	Erro_Euler All
0	10	0.200000	0.160000	7.096857e-20	1.925389e-08
1	20	0.100000	0.080000	1.560609e-37	2.450594e-14
2	30	0.066667	0.106667	1.050889e-53	2.650632e+61
3	40	0.050000	0.040000	1.297617e-60	9.903847e-26
4	50	0.040000	0.032000	2.351436e-66	2.563739e-31
5	100	0.020000	0.016000	7.558130e-68	2.149491e-59

Fonte: autores (2023)

A Figura 11 mostra os resultados para as simulações em que o passo h é menor que h_lim:

Figura 11 - Simulação com outros valores de k e estabilidade

	k	h_lim	h	Erro_Euler+Neural	Erro_Euler All
0	10	0.200000	0.140	6.663947e-20	3.069679e-14
1	20	0.100000	0.040	1.093237e-37	9.979987e-38
2	30	0.066667	0.020	2.696336e-55	2.480070e-55
3	40	0.050000	0.015	5.926980e-68	7.909295e-73
4	50	0.040000	0.016	7.427927e-79	3.062066e-90
5	100	0.020000	0.012	3.561543e-77	2.690511e-177

Fonte: autores (2023)

Nesse caso, o modelo Euler All apresenta menores valores de erro, quando comparados ao modelo Euler+Neural. Isso se deve ao fato que o método de Euler fornece bons resultados de aproximação quando o passo h é tomado cada vez menor.

5 CONSIDERAÇÕES FINAIS

O objetivo do artigo era a implementação de um modelo combinado do método de Euler e redes neurais para a aproximação da solução de um PVI. A implementação do modelo se mostrou eficaz principalmente nas situações em que o método de Euler perde a estabilidade, como mostrado na Figura 10. Em nossa análise, existe indicativo de uso do modelo construído nos casos em que o número de iterações (representado nas análises pelo parâmetro n_points) é fixo, e esse número faz com que o passo da iteração do método de Euler fique maior do que o limite necessário para garantir a estabilidade da solução.

REFERÊNCIAS

CHOI, R.Y *et al.* **Introduction to Machine Learning, Neural Networks, and Deep Learning.** Translational Vision Science & Technology, v.9, n.14, Jan. 2020.

BATHLA, G *et al.* **Autonomous Vehicles and Intelligent Automation: Applications, Challenges, and Opportunities.** Mobile Information Systems, v.2022, Jun.2022.

LEE, D.; YOON, S.N. **Application of Artificial Intelligence-Based Technologies in the Healthcare Industry: Opportunities and Challenges**. Int. J. Environ. Res. Public Health, v.18, n.1, Jan. 2021.

PATRÃO, M.; REIS, M. **Analisando a pandemia de COVID-19 através dos modelos SIR e SECIAR.** Biomatemática, v.30, 2022.

VIANA, M.; ESPINAR, J. **Equações Diferenciais: Uma abordagem de Sistemas Dinâmicos,** IMPA. 2021.

CRONIN, J. **Ordinary differential equations: introduction and qualitative theory.** CRC press, 2007.

BUTCHER, J. C. **Numerical methods for ordinary differential equations.** John Wiley & Sons, 2016.

CYBENKO, G. **Approximation by Superpositions of a Sigmoidal Function.** Math. Control Signals Systems, v2, Dec.1989.

ABU-MOSTAFA, Y. S.; MAGDON-ISMAIL, M.; LIN, H. **Learning from data**. AMLBook New York, 2012.

BURDEN, R. L.; FAIRES, D. J.; BURDEN, A. M. **Numerical analysis.** Cengage learning, 2015.