Proceeding Series of the Brazilian Society of Applied and Computational Mathematics, Vol. 3, N. 1, 2015.
Trabalho apresentado no XXXV CNMAC, Natal-RN, 2014.

Solução Numérica de Equações Diferenciais Usando Redes Neurais

Raul Barbosa Eluan, Alice Nabiça Moraes, Valcir João da Cunha Farias.
Faculdade de Estatística, ICEN, UFPA.
66000-000, Belém, PA
Email: rauleluan@gmail.com , alicenmoraes@gmail.com, valcir@ufpa.br

1. Introdução

Muitos problemas da Engenharia, Física e Biologia são modelados por equações diferenciais. Atualmente muitos métodos são utilizados para se obter aproximações de equações diferenciais, como por exemplo, Elementos Finitos, Diferenças Finitas, Método dos Momentos, dentre outros. Alguns desses métodos usam funções bases para representar a solução e, posteriormente, transformam o problema em um sistema linear. Outros aproximam as derivadas em alguns pontos selecionados, que constituem uma malha do domínio.

Neste trabalho são resolvidas, numericamente, equações diferenciais através de redes neurais artificiais do tipo feedforward [1,2]. Tal solução é apresentada através da soma de duas partes: a primeira é para satisfazer as condições iniciais e de contorno e a outra, é oriunda da rede neural com parâmetros ajustáveis (pesos e bias) de tal forma que minimiza uma função custo.

No treinamento da rede neural é empregado a técnica de otimização de Levenberg-Maquardt. Foi considerada uma rede neural com três camadas: a camada de entrada; uma camada oculta e a camada de saída. Na camada intermediária foram adotados dois neurônios e um neurônio na camada de saída.

2. Metodologia

Considere que $\Psi_t(\vec{x},\vec{p})$ seja a solução numérica de EDO/EDP com os parâmetros ajustáveis $\vec{p}$ (pesos e bias), o problema pode ser dado por:

$$\min_{\vec{p}} \sum_{x_i \in D} (G(\vec{x}, \Psi_t(\vec{x},\vec{p}), \nabla\Psi_t(\vec{x},\vec{p}), \nabla^2\Psi_t(\vec{x},\vec{p})))^2$$

Onde $G(\vec{x}, \Psi(\vec{x}), \nabla\Psi(\vec{x}), \nabla^2\Psi(\vec{x})) = 0$ é uma equação diferencial na forma geral, $\vec{x} = (x_1, \cdots, x_n) \in \Re^n$ e $D \subset \Re^n$. Com $\Psi_t(\vec{x}) = A(\vec{x}) + F(\vec{x}, N(\vec{x},\vec{p}))$ sendo $N(x,p)$ a saída da rede neural. Onde $A(\vec{x})$ satisfaz as condições iniciais e de contorno e não contém parâmetros ajustáveis; $N(\vec{x},\vec{p})$ é a saída da rede neural e F é construída para não contribuir com as condições iniciais/contorno.

3. Resultados

São apresentados dois exemplos de EDO, uma de 1º (1) e outra de 2º ordem (2), e um exemplo com EDP (3).

$$(1)\begin{cases} \dfrac{d\psi}{dx} + \psi = x + 2 \\ \{\psi(0) = 2 \quad 0 \le x \le 1 \end{cases}$$

$$(2)\begin{cases} \dfrac{d^2\psi}{dx^2} - \dfrac{\psi}{(x+1)} - 6 * \dfrac{d\psi}{dx} * \dfrac{1}{(3x+2)} = -x^2 + 2 \\ \psi(0) = 0 \\ \dfrac{d\psi(0)}{dx} = 0 \quad 0 \le \mathrm{x} \le 1 \end{cases}$$

$$(3)\begin{cases}\nabla^2\psi(x,y) = e^{-x}(x-2+y^3+6y)\\ \psi(0,y) = y^3;\\ \psi(1,y) = (1+y)e^{-1};\\ \psi(x,0) = xe^{-x};\\ \psi(x,1) = e^{-x}(x+1);\end{cases}$$

A Figura 1(a) apresenta a solução do problema (1) (EDO de 1º ordem), da qual observamos ter apresentado uma boa aproximação com um erro de apenas $3x10^{-3}$. A Figura 1(b) apresenta a solução obtida para o problema (2) (EDO 2º ordem) da qual observamos ter apresentado uma boa aproximação com um erro de apenas $4{,}5x10^{-3}$.

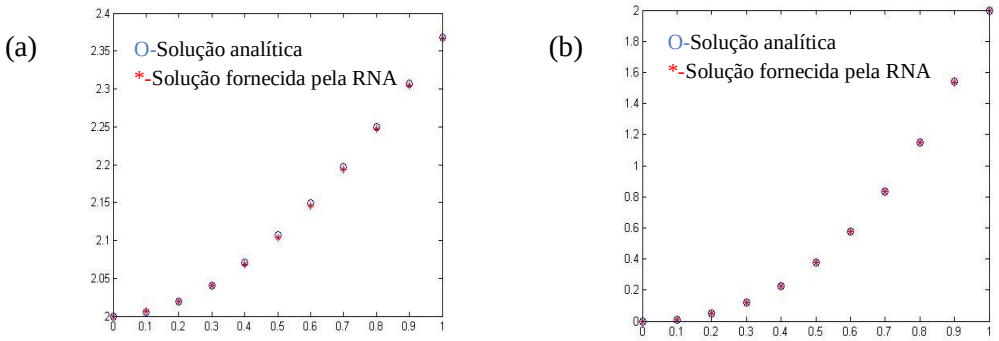

Figura 1 – (a) solução da EDO de 1º ordem. (b) Solução da EDO de 2º ordem.

A Figura 2 apresenta o erro da solução obtida pela rede neural multicamada com três com uma malha de 10 × 10 para o treinamento da rede. Como apresentou um erro máximo relativamente pequeno de $1.5x10^{-3}$, então podemos concluir que a solução fornecida pela rede apresentou uma boa solução em relação à solução analítica.

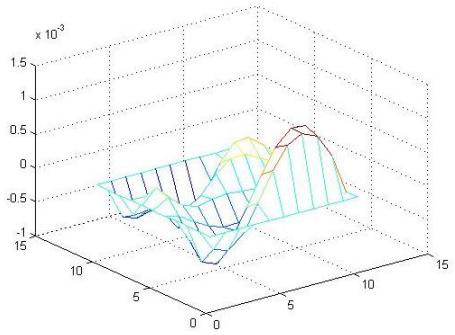

Figura 2 - Erro entre a solução fornecida pela rede e a solução analítica do problema (3).

Referências

[1] B. C. F. Batista, "Soluções de Equações Diferenciais Usando Redes Neurais de Múltiplas camadas com os métodos da Descida mais íngreme e Levenberg-Marquardt". Dissertação de mestrado, PPGME-ICEN-UFPA, 2012.

[2] I. E. L. Aristidis Likas, Artificial Neural Networks for Solving Ordinary and Partial Differential Equations. *IEEE Transactions on Neural Networks*, 9(1998), 987-1000.

[3] H. Lee and I. Kang, Neural algorithms for solving differential equations, *J. Comput. Phys.*, vol. 91(1990), 110–117.

