



## **REDES NEURAIIS E O ESTUDO DE SÉRIES TEMPORAIS: UMA APLICAÇÃO NO ENSINO DE ENGENHARIA**

**José Tarcísio Franco de Camargo** – [jtfc@bol.com.br](mailto:jtfc@bol.com.br)  
Faculdade Municipal "Professor Franco Montoro", FMPFM  
Centro Regional Universitário de Espírito Santo do Pinhal, UNIPINHAL  
Rua dos Estudantes s/n, Cachoeira de Cima  
CEP 13840-000 – Mogi Guaçu - SP  
**Estéfano Vizconde Veraszto** – [estefanovv@cca.ufscar.br](mailto:estefanovv@cca.ufscar.br)  
Universidade Federal de São Carlos  
Depto. de Ciências da Natureza, Matemática e Educação, UFSCar, CCA  
Rodovia Anhanguera, Km 174  
CEP 13604-900 – Araras - SP

**Resumo:** *As séries temporais constituem um objeto de estudo muito importante nas áreas das ciências, engenharias e negócios, especialmente em casos onde espera-se conhecer, prever e otimizar comportamentos. Neste contexto, pretende-se mostrar a viabilidade do uso das redes neurais artificiais no estudo de diversas séries temporais presentes em um curso de engenharia, em especial aquelas que não possuem um comportamento explícito ou não são passíveis de serem modeladas matematicamente de forma simples e que tenham aplicação direta na formação de futuros engenheiros.*

**Palavras-chave:** *ensino de engenharia, séries temporais, modelagem matemática*

### **1. INTRODUÇÃO**

Conhecer, prever e otimizar comportamentos são objetivos bastante comuns no estudo das séries temporais. Ser capaz de analisar um conjunto de dados, entender o seu comportamento e fazer previsões aproximadas para valores futuros pode-se tornar um fator crucial para o sucesso de um empreendimento. Dessa forma, o estudo das séries que não apresentam um comportamento de fácil modelagem, tais como as provenientes de áreas como *data mining*, otimização de processos ou previsão de demandas é, portanto, um objetivo a ser tratado com cuidado nos cursos de engenharia. Os futuros engenheiros devem, assim, estar aptos a analisar e “extrair conhecimento” de séries temporais de modelagem não trivial, sendo capazes de, na medida do possível, gerar previsões para situações futuras. Neste contexto, este artigo apresenta uma metodologia, o uso das redes neurais para o estudo das séries temporais, que pretende contribuir para que esse objetivo seja alcançado.

### **2. SÉRIES TEMPORAIS NO CONTEXTO DO ENSINO DE ENGENHARIA**

O estudo das séries temporais pode ser realizado com maior frequência através do uso de equações bem definidas, quando a série apresenta um comportamento bem determinado, ou através de equações de aproximação ou interpolação, quando existe um comportamento que pode ser identificado, mas não modelado com exatidão.

Por exemplo, a série temporal apresentada na Tabela 1 pode ser perfeitamente definida através da equação:

$$x_{k+1} = x_k + x_{k-1} \quad (1)$$

sendo  $x_0 = x_1 = 1$

Tabela 1 - Série de Fibonacci.

1	1	2	3	5	8	13	21	34	55	89	144	...
---	---	---	---	---	---	----	----	----	----	----	-----	-----

A série temporal indicada pela Tabela 2, que representa a evolução da população brasileira no período entre 1900 e 1990, pode ser aproximada por uma função exponencial do tipo:

$$y = a_0 \cdot e^{a_1 \cdot x} \quad (2)$$

onde  $a_0$  e  $a_1$  são constantes que podem ser determinadas através da solução do sistema:

$$[X]^T \cdot [X] \cdot [A] = [X]^T \cdot [Y] \quad (3)$$

Tabela 2 - População brasileira no decorrer do Século XX. (IBGE, 2011).

Ano (x)	1900	1910	1920	1930	1940	1950	1960	1970	1980	1990
Pop. em milhões (y)	17,4	23,4	30,6	35,8	41,2	51,9	70,9	94,5	121,1	146,9

Embora os casos acima apresentem um comportamento relativamente previsível, deve-se considerar um problema a ser contornado: nem sempre uma série temporal possui um comportamento bem definido, que possa ser descrito de forma exata através de uma equação, bem como nem sempre temos as séries a serem estudadas apresentando um comportamento que permita uma aproximação ou interpolação através de uma determinada equação matemática. É neste contexto que pretende-se apresentar o uso das redes neurais no estudo das séries temporais: estas podem detectar e aprender comportamentos que não são modelados matematicamente de forma trivial.

Conforme apresentado em Lippmann (1987), pode-se afirmar que as Redes Neurais Artificiais (RNAs) são capazes de identificar comportamentos a partir do aprendizado prévio de determinados padrões. Assim, os comportamentos das séries apresentadas acima podem ser aprendidos por uma determinada RNA, tornando possível inferir valores para esta série, dentro de certos limites. Além disso, comportamentos implícitos, que não poderiam ser observados de maneira direta ou de forma trivial, podem vir a ser detectados através de uma RNA. Por exemplo, o estudo de uma série de preços de uma determinada ação negociada na bolsa de valores pode ser realizado através de uma RNA. Neste sentido, a Tabela 3 apresenta uma série temporal composta pelas médias móveis aritméticas de cinco dias do preço de fechamento da ação EMBR3 (EMBRAER – ON) negociada na Bolsa de Valores de São Paulo (BOVESPA).

Tabela 3 - Média móvel aritmética dos preços de fechamento da ação EMBR3.

Data (x)	14/03/13	15/03/13	18/03/13	19/03/13	20/03/13	21/03/13	22/03/13	...
Preço em Reais (y)	17,22	17,25	17,27	17,28	17,28	17,33	17,40	...

Um modelo de RNA bastante interessante para o estudo das séries temporais é o perceptron em múltiplas camadas (MLP – multilayer perceptron). Através deste modelo, padrões implícitos em uma série temporal podem vir a ser detectados, permitindo que determinados comportamentos possam vir a ser identificados pela rede. Assim, a análise de séries temporais aqui proposta encontra-se focada no uso de RNAs do tipo MLP, como alternativa aos métodos convencionais para o estudo das séries.

### 3. REDES NEURAIS E ENSINO DE ENGENHARIA

Uma RNA constitui um sistema, comumente abstrato, que pode ser utilizado em várias aplicações, tais como reconhecimento e classificação de padrões, modelagem de problemas físicos e/ou matemáticos, controle de processos diversos e processamento de sinais, bem como diversas outras aplicações tais como apresentado em Philip et al. (2011), Tarsauliya et. al. (2010), Burke (1995), Mighell (1988), Reategui (1994) e Yoda (1994). Para tanto, antes de seu uso nas tarefas mencionadas, a RNA deve ser “treinada” através de exemplos de padrões de comportamento. Assim, em sua fase de treinamento, a rede é apresentada a diversos padrões de entrada e seus respectivos padrões de saída desejados, de forma a “aprender” a “lei de formação” que correlaciona cada padrão de entrada ao seu respectivo padrão de saída.

Após esta fase inicial, espera-se que, caso seja apresentado um padrão de entrada não pertencente ao conjunto de treinamento, a rede seja capaz de inferir um provável padrão de saída. Como exemplo, considere que deseja-se verificar uma possível lei de formação para a série descrita na Tabela 3.

Neste caso, durante a sua fase de treinamento, poderia ser apresentado à rede o seguinte conjunto de padrões de entrada e saída, para que a mesma pudesse vir a compreender a lei de formação desta série:

Tabela 4 – Conjuntos de treinamento para uma RNA.

Vetor de entrada	Saída desejada
17,22 ; 17,25 ; 17,27 ; 17,28	17,28
17,25 ; 17,27 ; 17,28 ; 17,28	17,33
17,27 ; 17,28 ; 17,28 ; 17,33	17,40

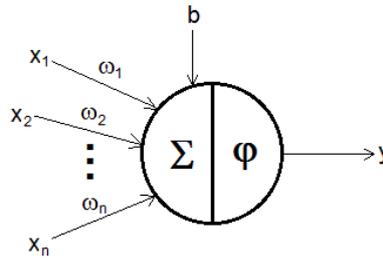
Assim, após a fase de treinamento, caso venha a ser apresentado à rede o padrão de entrada (17,27 ; 17,28 ; 17,28 ; 17,33), espera-se que a mesma apresente em sua saída uma estimativa para a média móvel em 22/03/13.

Um modelo de RNA que se encaixa dentro desta proposta é o MLP (HAYKIN, 2001), o qual é discutido a seguir.

#### 3.1. O perceptron

Uma RNA do tipo MLP é constituída pela organização em camadas do elemento computacional denominado “perceptron”. Um perceptron, por sua vez, caracteriza-se por tentar expressar uma representação simbólica (e matemática) de um neurônio biológico. A Figura 1 apresenta o modelo simbólico de um perceptron. Seu modelo é amplamente discutido em Lippmann (1987), Hush & Horne (1993) e Haykin (2001), sendo resumidamente apresentado a seguir. Günther & Fritsch (2010), Hush & Horne (1993), Haykin (2001) e Fahlman (1988) também apresentam detalhadamente o “algoritmo de retropropagação” (backpropagation) que será discutido mais adiante.

Figura 1 - Modelo simbólico de um perceptron ou neurônio artificial.



Neste modelo temos um “nó”, que representa o corpo do neurônio, alimentado por diversos estímulos externos ( $\mathbf{x}$ ) através de “conexões sinápticas”. O resultado da computação dos estímulos externos pelo perceptron é exportado através de uma conexão sináptica de saída ( $y$ ) que encaminhará este sinal para a entrada de outros perceptrons da rede.

Os estímulos externos recebidos por um perceptron podem ser provenientes das saídas exportadas por outros perceptrons ou provenientes da entrada da RNA ou, então, provenientes de um “sinal de polarização” (“bias”) particular do próprio neurônio. Segundo German (1992), o sinal de polarização possui grande importância no controle de ruídos dos dados apresentados à rede. Todos os sinais que chegam a um neurônio são ponderados pelo “peso sináptico” da conexão que leva este sinal ao neurônio.

O equacionamento matemático que formaliza este modelo leva em consideração que os estímulos de entrada tem seus efeitos somados de forma ponderada. Ou seja:

$$v = \sum_{i=1}^n \omega_i \cdot x_i + b \quad (4)$$

onde:

- “ $v$ ” é a soma ponderada dos estímulos de entrada
  - “ $w_i \cdot x_i$ ” é o produto entre um sinal de entrada “ $x$ ” e o peso sináptico “ $w$ ” da conexão que encaminha o sinal ao neurônio
- “ $b$ ” é o sinal de polarização particular deste neurônio

A resposta do perceptron aos estímulos recebidos é função da soma ponderada destes. Ou seja:

$$y = \varphi(v) \quad (5)$$

onde:

- “ $y$ ” é a resposta do neurônio
- $\varphi(v)$  é a “função de ativação” do neurônio, que correlaciona a saída aos estímulos de entrada.

A função de ativação “ $\varphi(v)$ ” do perceptron deve ser não-linear, de forma a permitir que a RNA como um todo possa representar funções não-lineares, além de ser contínua e “suave”, de forma a ser diferenciável em todo o seu intervalo de consideração.

Um exemplo de função de ativação bastante popular em redes MLP é a função tangente-hiperbólica:

$$\varphi(v) = \alpha \cdot \tanh(\beta \cdot v) \quad (6)$$

onde:

“ $\alpha$ ” e “ $\beta$ ” são constantes quaisquer.

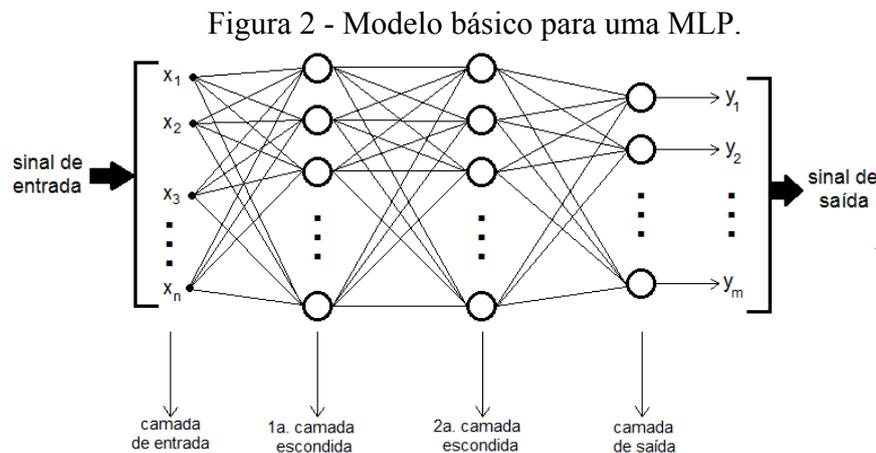
Para esta função de ativação, LeCun (1989) considera  $\alpha = 1,7159$  e  $\beta = 2/3$  como valores adequados.

### 3.2. RNAs em camada

Conforme mencionado previamente, uma RNA do tipo MLP é composta pela associação em camadas de perceptrons. A Figura 2 mostra um modelo para uma MLP.

Nesta arquitetura temos uma “camada de entrada”, onde são aplicados os estímulos (sinais) de entrada da rede, que são propagados camada por camada até a saída da rede, desde a “1ª. camada escondida” de neurônios até a “camada de saída” da rede, que apresentará o resultado de todo o processamento da rede ao meio externo.

Também deve ser notado que em uma MLP as entradas de um determinado neurônio estão conectadas às saídas de todos os neurônios da camada imediatamente anterior e, a saída deste neurônio, está conectada a todas as entradas dos neurônios da camada imediatamente posterior.



### 3.3. O aprendizado de uma RNA - O algoritmo de retropropagação

Durante a fase de aprendizado, para que a rede possa aprender o comportamento desejado, deve ser utilizado um algoritmo recursivo como o “algoritmo de retropropagação de erro” (ou error backpropagation).

No algoritmo de retropropagação, durante a fase de treinamento da rede, um padrão de entrada “x” é apresentado à rede, sendo propagado em direção à saída, gerando nesta um padrão “o”, o qual é comparado com o valor desejado “d” para o padrão de entrada em questão.

A diferença “d-o” entre o padrão desejado e o obtido de fato na saída da rede é o erro “e” desta fase de aprendizado para este padrão. Se  $e=0$ , então conclui-se que a rede “aprendeu” o padrão de entrada “x”. Se  $e \neq 0$ , então o erro deve ser retro-propagado (da camada de saída em direção à 1ª. camada escondida) de forma a ajustar os pesos das conexões entre os neurônios das camadas.

Formalmente, temos que:

$$\text{sinal de erro:} \quad e_j = d_j - o_j$$

onde:

- “ $d_j$ ” é um elemento do vetor de saída desejado para o padrão de entrada apresentado.



- “ $o_j$ ” é um elemento do vetor de saída obtido com a propagação do vetor de entrada.
- $e_j$ ” é um elemento do vetor de erro obtido.

Conforme Andrews e Geva (1994), para minimizar as dificuldades de treinamento de uma RNA a função de erro deve ser não linear. Assim, para o padrão de entrada apresentado, definimos a soma instantânea dos erros quadráticos na saída da rede por:

$$\varepsilon = \frac{1}{2} \cdot \sum_{j=1}^m e_j^2 \quad (7)$$

onde:

- “ $m$ ” é o número de elementos do vetor de erro (que equivale ao número de neurônios na camada de saída da rede).

Se considerarmos todos os padrões de entrada a serem apresentados na fase de treinamento, poderemos determinar o “erro médio quadrático” para estes padrões durante uma “época” (dá-se o nome de “época” a cada propagação/retropropagação realizada para todos os padrões de entrada apresentados à rede em sua fase de treinamento). Assim:

$$\varepsilon_{QM} = \frac{1}{N} \cdot \sum_{n=1}^N \varepsilon(n) \quad (8)$$

onde:

- “ $N$ ” é o número de padrões de treinamento apresentados à rede
- “ $\varepsilon(n)$ ” é o erro obtido na apresentação do padrão “ $n$ ”

Portanto, tanto menos será o valor do erro médio quadrático ( $\varepsilon_{QM}$ ) quanto melhor for o aprendizado dos padrões de entrada pela rede em uma determinada época. Dessa forma, se  $\varepsilon_{QM} = 0$ , a rede aprendeu com precisão absoluta todos os padrões apresentados.

Assim, uma proposta para treinamento de uma MLP pode ser:

- Enquanto  $\varepsilon_{QM}$  não for suficientemente baixo:
  - a) Para cada padrão de treinamento:
    - i. propagar este padrão, em direção à saída, camada por camada
    - ii. calcular o erro “ $e$ ” entre a saída desejada e a saída obtida
    - iii. Retropropagar o erro, da saída em direção à entrada, corrigindo os pesos sinápticos de cada conexão.
    - iv. Fim
  - b) Atualizar o valor de  $\varepsilon_{QM}$
- Fim

No algoritmo mencionado acima, a correção do erro para ajuste dos pesos sinápticos pode ser realizada através da “regra delta”. Este ajuste nos pesos sinápticos é fundamental para o aprendizado da RNA, uma vez que o conhecimento adquirido pela rede é condensado nestes. Considerando a conexão existente entre dois neurônios de camadas consecutivas, a atualização do peso sináptico “ $\omega_{ji}$ ” entre estes neurônios, de uma iteração para outra, será dada por:

$$\omega_{ji}(t+1) = \omega_{ji}(t) + \Delta\omega_{ji} \quad (9)$$



onde:

- “ $\omega_{ji}(t)$ ” é o valor atual do peso sináptico
- “ $\Delta\omega_{ji}$ ” é a correção do peso sináptico a ser aplicada
- “ $\omega_{ji}(t+1)$ ” é o valor atualizado (próximo valor) do peso sináptico

O cálculo da correção do valor do peso sináptico ( $\omega_{ji}$ ) deve vir na direção contrária ao gradiente do erro acumulado na propagação de uma época das amostras de entrada (conjunto de treinamento). Assim, de acordo com a regra delta, a correção do peso sináptico pode ser dada por:

$$\Delta\omega_{ji} = \eta \cdot \delta_j \cdot y_i \quad (10)$$

onde:

- “ $\eta$ ” é a “taxa de aprendizado” do neurônio ( $0 < \eta < 1$ ), cujo valor, arbitrado pelo usuário da rede, define quão rapidamente a rede deverá convergir para o ponto de mínimo.
- “ $\delta_j$ ” é o “gradiente local” do neurônio “j” para a correção do erro, calculado em função do erro retropropagado pela rede.
- “ $y_i$ ” é o valor da saída do neurônio “i”.

O valor do gradiente local “ $\delta_j$ ” para o neurônio “j” pode ser calculado da seguinte forma:

$$\delta_j = \varphi' \cdot e_j, \text{ se o neurônio “j” pertence à camada de saída;}$$

ou

$$\delta_j = \varphi' \cdot \sum_k \delta_k \cdot \omega_{kj}, \text{ Se o neurônio “j” pertence a uma camada escondida.}$$

De forma a melhorar a convergência da regra delta, pode ser utilizado um “fator de momento” (“ $\mu$ ”) que irá considerar o valor de  $\Delta\omega_{ji}$  da iteração anterior para o cálculo do próximo valor:

$$\Delta\omega_{ji}(t) = \mu \cdot \Delta\omega_{ji}(t-1) + \eta \cdot \delta_j(t) \cdot y_i(t) \quad (11)$$

sendo o fator de momento “ $\mu$ ” arbitrado entre 0 e 1.

De forma simplificada, o processo de criação e treinamento de uma RNA pode ser descrito na forma:

#### 1. Criação da rede:

- Definir o tamanho do vetor de entrada da rede, ou seja, o número de nós da camada de entrada da rede.
- Definir o tamanho do vetor de saída da rede, ou seja, o número de neurônios da camada de saída da rede.
- Definir o número de camadas escondidas da rede.
- Definir o número de neurônios de cada camada escondida da rede.
- Atribuir valores aleatórios (entre -1 e 1) para os sinais de polarização de cada neurônio.
- Atribuir valores aleatórios (entre -1 e 1) para os pesos sinápticos de cada conexão da RNA.
- Definir a função de ativação dos neurônios e sua derivada.

#### 2. Treinamento da rede:

- Definir os valores para “ $\eta$ ” e “ $\mu$ ”.
- Definir o valor limite para  $\epsilon_{MQ}$ .



- c. Enquanto  $\epsilon_{MQ}$  não for suficientemente baixo:
- i. Para cada padrão de treinamento:
    1. Aplicar o padrão à entrada “x” da RNA.
    2. Calcular a saída “y” de cada neurônio da 1ª. camada escondida, aplicando estas saídas às entradas da camada subsequente, calculando também as saídas desta camada até a camada de saída (propagação do sinal de entrada).
    3. Calcular o erro “e” para este padrão, comparando o valor desejado para a saída (“d”) e o valor obtido na saída (“o”) a partir da entrada aplicada.
    4. A partir da camada de saída, em direção à camada de entrada da rede (retropropagação do erro):
      - a. Determine o gradiente local (“ $\delta$ ”) de cada neurônio de cada camada.
      5. A partir da camada de saída, em direção à camada de entrada da rede:
        - a. Determine  $\Delta\omega$  de cada conexão sináptica.
        - b. Atualize  $\omega$  de cada conexão sináptica.
    6. Fim.
  - ii. Atualize o valor de  $\epsilon_{MQ}$  para esta época.
  - iii. Fim.

Após o treinamento da rede, o uso desta é realizado simplesmente pela apresentação de um padrão de entrada qualquer, o qual é propagado em direção à saída da rede, onde poderá ser observada a resposta da rede ao estímulo de entrada apresentado.

#### 4. IMPLEMENTAÇÃO SOBRE CASOS DE ESTUDO

A implementação e a simulação de uma RNA do tipo MLP pode ser realizada através do desenvolvimento de programas de computador em diversas linguagens. Estes autores realizaram a implementação de RNAs na linguagem de programação do “SciLab” (<http://www.scilab.org>), obtendo os seguintes resultados para as simulações realizadas.

De acordo com Cybenko (1989) e Hertz (1991), uma RNA com uma única camada intermediária é suficiente para aproximar qualquer função contínua. Por sua vez, Cybenko (1988) afirma que qualquer função matemática pode ser modelada através de uma RNA com, no máximo, duas camadas intermediárias. Funuhashi (1989) também investigou o número de camadas intermediárias necessárias para a implementação de classes de funções em uma RNA. Assim, como as séries que pretendemos simular podem ser aproximadas por funções contínuas, todas as simulações foram realizadas com RNAs dotadas de uma camada escondida. De fato, nas simulações indicadas a seguir foi utilizada uma única RNA com quatro neurônios na camada de entrada, uma camada escondida com cinco neurônios e um neurônio camada de saída. Em todos os casos também foram utilizados os mesmos parâmetros de treinamento: taxa de aprendizado ( $\eta$ ) igual a 0,3, fator de momento ( $\mu$ ) igual a 0,5 e função de ativação dos neurônios do tipo tangente hiperbólica.

A Tabela 5 apresenta os resultados para a simulação da Série de Fibonacci, apresentada inicialmente na Tabela 1.

Na Série de Fibonacci a razão  $x_{k+1}/x_k$  converge para 1,618. Em nossa simulação, nota-se uma convergência para 1,612 no extremo direito da Tabela 5. Em média, verifica-se que esta razão, para os dados da simulação, gira em torno de 1,641, o que indica um aprendizado razoável desta série pela RNA.

Tabela 5 - Simulação da Série de Fibonacci em uma RNA.

Valor Real	5	8	13	21	34	55	89	144	233	377	610
Valor Simulado	4.3	7.0	11.8	19.4	31.9	52.1	85.3	139.9	229.9	377.2	608.2

Os resultados para uma simulação da evolução da população brasileira no Século XX são apresentados na Tabela 6.

Tabela 6 - Simulação para a população brasileira no decorrer do Século XX através de uma RNA.

Ano	1940	1950	1960	1970	1980	1990
Dado do IBGE (em milhões)	41,2	51,9	70,9	94,5	121,1	146,9
Valor simulado (em milhões)	40,9	50,8	70,0	94,0	118,0	146,6

Pode-se verificar que o valor simulado encontra-se muito próximo dos valores reais apresentados pelo IBGE, sendo observado um erro de, no máximo, 2,5%. Quanto solicitada a inferir informações para os anos 2000 e 2010 a rede apresentou os valores indicados na Tabela 7, os quais são comparados com os dados apresentados pelo IBGE.

Tabela 7 - Resultados sobre a evolução da população brasileira inferidos pela rede.

Ano	2000	2010
Dado oficial (IBGE, 2011)	169,6 milhões	190,8 milhões
Resultado inferido pela rede	181,3 milhões	227,4 milhões

Pode ser observado que o valor inferido para o ano 2000 apresenta uma aproximação razoável em relação ao valor real (em torno de 6,5% do valor verificado de fato). Por sua vez, o resultado inferido para o ano 2010 pode ser parcialmente explicado por uma redução na tendência de crescimento da população brasileira, verificada a partir do final do Século XX, a qual mudou o comportamento da série estudada.

A Tabela 8 apresenta uma comparação entre as médias móveis aritméticas (m.m.a.) efetivamente calculadas e as simuladas pela RNA para a ação EMBR3.

Tabela 8 - Média móvel aritmética dos preços de fechamento da ação EMBR3.

Data	14/03/13	15/03/13	18/03/13	19/03/13	20/03/13	21/03/13	22/03/13	...
m.m.a. verificada	17,22	17,25	17,27	17,28	17,28	17,33	17,40	...
m.m.a. simulada	17,20	17,34	17,41	17,43	17,45	17,44	17,45	

Nota-se, para a Tabela 8, uma diferença inferior a 1% entre os valores verificados e simulados. Quando solicitada a inferir a m.m.a. para os dias entre 25/03/13 e 27/03/13 a rede apresentou os seguintes resultados, que são comparados com os valores verificados na Tabela 9.



Tabela 9 - Comparativo entre valores inferidos e verificados pela rede, para o período entre 25/03/13 e 27/03/13.

Data	25/03/13	26/03/13	27/03/13
m.m.a. verificada	17,37	17,53	17,58
m.m.a. inferida	17,54	17,58	17,63

Através da observação dos valores apresentados, pode-se notar que, mesmo neste caso, o erro entre os valores verificados e inferidos também é inferior a 1%.

## 5. CONSIDERAÇÕES FINAIS

Este trabalho apresentou uma alternativa ao estudo e ensino das séries temporárias através do uso das RNAs. Ao longo do texto o modelo das RNAs do tipo MLP foi discutido e exemplos de aplicações foram apresentados.

Através deste estudo pode-se constatar que a aplicação das RNAs sobre as séries temporais que apresentam um comportamento perfeitamente definido através de uma equação, como no caso da Série de Fibonacci, não produz efeito prático, uma vez que os resultados da simulação de uma RNA serão sempre aproximações de valores que podem ser perfeitamente determinados. Por sua vez, com relação às séries cujo equacionamento matemático pode ser apenas aproximado, como no caso da série que representa a evolução da população brasileira, o uso das RNAs pode constituir uma ferramenta de análise bastante interessante, através da qual comportamentos implícitos podem vir a ser detectados. Com relação às séries de natureza mais complexa, como no caso das séries temporais de preços de ações, o uso das RNAs pode se constituir em uma ferramenta fundamental de análise, apontando comportamentos que dificilmente poderiam vir a ser detectados através de outros métodos.

O grande inconveniente no uso das RNAs para a análise das séries temporais encontra-se na dificuldade de se encontrar a topologia de rede mais adequada para cada caso a ser simulado. Utilizar um número grande de camadas intermediárias não é recomendado, pois cada vez que o erro medido durante o treinamento é propagado para a camada anterior ele pode vir a se tornar mais elevado. Mesmo respeitadas as condições de modelagem apresentadas em Cybenko (1989), Hertz (1991) e Cybenko (1988), onde sugere-se o uso de não mais do que duas camadas intermediárias para qualquer função matemática, a definição do número de neurônios em cada camada escondida não é uma tarefa simples.

O número de nodos nas camadas intermediárias de uma RNA depende fortemente da distribuição dos padrões de treinamento e validação da rede. A utilização de muitas unidades pode facilitar a memorização dos padrões de treinamento. Contudo, isso limitará a capacidade de extrair as características gerais que permitem a generalização ou reconhecimento de padrões não vistos durante o treinamento (este problema é chamado de overfitting). Por outro lado, um número muito pequeno de nodos pode forçar a rede a gastar tempo em excesso tentando encontrar uma representação ótima. Se o número de exemplos for muito maior que o número de conexões entre nodos o overfitting é improvável, mas pode ocorrer underfitting (a rede não converge durante o seu treinamento).

Finalmente, a aplicabilidade desta proposta em sala de aula justifica-se pela necessidade de fornecer aos alunos dos cursos de engenharia novas ferramentas para o estudo de casos que não são necessariamente cobertos pelas ferramentas tradicionais de



ensino/aprendizagem (*forecasting*, por exemplo). Neste sentido, o uso das redes neurais no ensino das séries temporais também procura instigar os alunos a pesquisarem novas e melhores ferramentas para a solução de problemas do seu cotidiano, considerando-se, assim, a indissociabilidade entre ensino e pesquisa em um curso de graduação. Especificamente no caso destes autores, a metodologia apresentada ao longo deste texto foi inicialmente introduzida na disciplina “Cálculo Numérico”, do sexto semestre de um Curso de Ciência da Computação. Nesta disciplina, os alunos tiveram a oportunidade de estudar o tema das séries temporais e desenvolver programas de computador direcionados a este assunto, com vistas à solução de determinados problemas. As RNAs foram superficialmente apresentadas como possível ferramenta para a solução de problemas associados a séries temporais nesta disciplina. Por sua vez, quando os alunos que cursaram esta disciplina passaram a frequentar a disciplina “Inteligência Artificial” (no decorrer do oitavo semestre do Curso de Ciência da Computação), a proposta aqui apresentada foi tratada com grande ênfase, procurando fomentar entre os alunos, além do uso propriamente desta metodologia, a necessidade do constante desenvolvimento de pesquisas como forma de continuamente buscar o aperfeiçoamento de soluções em problemas de engenharia.

## REFERÊNCIAS BIBLIOGRÁFICAS

- ANDREWS, R.; GEVA, S. Rule extraction from a constrained error backpropagation MLP. In Proceedings of the 5th Australian Conference on Neural Networks, pp. 9-12, Brisbane, Austrália, 1994.
- BURKE, H.; ROSEN, D.; GOODMAN, P. Comparing the prediction accuracy of artificial neural networks and other statistical models for breast cancer survival. In G. TESAURO, D. S. TOURETZKY, T. K. LEEN, editors, Neural Information Processing Systems 7. MIT Press, 1995.
- CYBENKO, G. Continuous valued neural networks with two hidden layers are sufficient. Technical report, Department of Computer Science, Tufts University, 1988.
- CYBENKO, G. Approximation by superpositions of a sigmoid function. *Mathematics of Control, Signals and Systems*, 2:303-314, 1989.
- FAHLMAN, S.E. An empirical study of learning speed in backpropagation networks. Technical report, Carnegie Mellon University, 1988.
- FUNUHASHI, K. I. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2:183-192, 1989.
- GERMAN, S.; BIENESTOCK, E.; DOURSAT, R. Neural networks and the bias-variance dilemma. *Neural Computation*, 4:1-58, 1992.
- GÜNTHER, F.; FRITSCH, S. Neuralnet: Training of Neural Networks. *The R Journal*, 2/1: 30-38, 2010. (ISSN: 2073-4859)
- HAYKIN, S. 2001. *Redes Neurais – Princípios e Prática*, 2ª. Ed. Porto Alegre: Bookman, 2001. (ISBN: 978-85-7307-718-6)



HERTZ, J.; KROGH, A.; PALMER, R.G. Introduction to the Theory of Neural Computation, volume Lecture Notes. Vol. 1 of Santa Fe Institute Studies in The Science of Complexity. Addison-Wesley, 1991.

HUSH, D. R. & HORNE, B. G. Progress in Supervised Neural Networks – What’s New Since Lippmann?, IEEE Signal Processing Magazine, January/1993, pp. 8-39.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA (IBGE). Sinopse do Censo Demográfico 2010. Rio de Janeiro: IBGE, 2011. Disponível em: <[http://www.ibge.gov.br/home/estatistica/populacao/censo2010/tabelas\\_pdf/Brasil\\_tab\\_1\\_4.pdf](http://www.ibge.gov.br/home/estatistica/populacao/censo2010/tabelas_pdf/Brasil_tab_1_4.pdf)>. Acessado em 22/05/2013.

LeCUN, Y. Generalization and network design strategies. Technical Report CRG-TR-89-4, Department of Computer Science, University of Toronto, Canada, 1989.

LIPPMANN, R. P. An Introduction to Computing with Neural Nets. IEEE ASSP Magazine, April/1987, pp. 4-22.

MIGHELL, D.A.; WIKINSON, T.S.; GOODMAN, J.W. Back propagations and its application to handwritten signature verification. In R.P. LIPPMANN, J.E. MODDY, D.S. TOURETZKY, editors, Advances in Neural Information Processing Systems 2. Morgan Kaufmann, 1988.

PHILIP, A.A.; TAOFIKI, A.A.; BIDEMI, A.A. Artificial Neural Network Model for Forecasting Foreign Exchange Rate. World of Computer Science and Information Technology Journal (WCSIT), v.1, n.3, pp. 110-118, 2011.

REATEGUI, E.; CAMPBELL, J.A. A classification system for credit card transactions. In Proceedings of the Second European Workshop on Case-Based Reasoning, pp.167-174, November 1994.

TARSAULIYA, A.; KANT, S.; KALA, R.; TIWARI, R.; SHUKLA, A. Analysis of Artificial Neural Network for Financial Time Series Forecasting. International Journal of Computer Applications, v.9, n.5, pp. 16-22, 2010. (ISSN: 0975-8887)

YODA, M. Predicting the Tokyo stock market. John Wiley & Sons, 1994.

## **NEURAL NETWORKS AND THE STUDY OF TIME SERIES: AN APPLICATION IN ENGINEERING EDUCATION**

**Abstract:** The time series are an important object of study in the sciences, engineering and business, especially in cases where it is expected to know, predict and optimize behaviors. In this context, we intend to show the feasibility of using artificial neural networks in the study of several time series present in an engineering course, especially those who have no overt behavior or are not able to be modeled mathematically in a simple way and have direct application in the education of future engineers.

**Key-words:** engineering education, time series, mathematical modeling