

## Previsão de séries temporais com a utilização redes neurais artificiais

### Time series forecasting using artificial neural networks

Victor Cesar Simões de Oliveira\*,  
Cristiellen Heuko da Silva†, Hugo Valadares Siqueira‡

#### RESUMO

Séries temporais são conjuntos de dados que evoluem ao longo do tempo e são encontradas em diversas áreas, desde finanças até medicina e meio ambiente. O domínio da previsão de séries temporais é fundamental para a tomada de decisões em cenários complexos e dinâmicos. Este trabalho apresenta uma abordagem eficaz para a previsão de dados de séries temporais por meio da utilização de Redes Neurais Artificiais (RNAs), com foco na arquitetura de Perceptron de Múltiplas Camadas (MLP). O trabalho descreve os princípios fundamentais das RNAs e da arquitetura MLP, incluindo detalhes sobre a função de ativação, métrica de erros e a importância da divisão de dados em conjuntos de treinamento, validação e teste. Em seguida, apresenta-se a metodologia específica utilizada para aplicar a MLP na previsão de séries temporais. Os resultados obtidos a partir da aplicação da MLP em diferentes séries temporais são discutidos, demonstrando a capacidade da rede neural em prever com precisão os valores futuros. Além disso, são abordadas considerações sobre o desempenho do modelo, incluindo métricas de avaliação, como o erro quadrático médio (MSE, do inglês *Mean Squared Error*).

**PALAVRAS-CHAVE:** séries-temporais; previsão; redes-neurais-artificiais; MLP.

#### ABSTRACT

Time series are sets of data that evolve over time and are found in various fields, from finance to medicine and the environment. The domain of time series forecasting is crucial for decision-making in complex and dynamic scenarios. This work presents an effective approach to forecasting time series data using Artificial Neural Networks (ANNs), with a focus on the Multilayer Perceptron (MLP) architecture. The work describes the fundamental principles of ANNs and the MLP architecture, including details about activation functions, error metrics, and the importance of dividing data into training, validation, and test sets. Next, the specific methodology used to apply MLP in time series forecasting is presented. The results obtained from applying MLP to different time series are discussed, demonstrating the neural network's ability to accurately predict future values. Additionally, considerations about model performance are addressed, including evaluation metrics such as Mean Squared Error (MSE).

**KEYWORDS:** time-series; prevision; artificial-neural-networks; MLP.

## INTRODUÇÃO

Uma Rede Neural Artificial (RNA) é um modelo computacional inspirado no sistema nervoso dos organismos superiores. Assim como o cérebro humano é composto por bilhões de neurônios, uma RNA é formada por unidades de processamento interconectadas chamadas neurônios artificiais. Ambos os sistemas processam informações e as transmitem para outros neurônios através de conexões entre tais unidades, conhecidas como sinapses. Podem realizar operações complexas através da interação entre essas unidades de processamento (SILVA; SPATTI; FLAUZINO, 2010).

\* Bolsista PIBIT/CNPQ.  Universidade Tecnológica Federal do Paraná, Ponta Grossa, Paraná, Brasil. ✉ vic-torcesar@alunos.utfpr.edu.br. ☎ 2830129570630114.  0009-0003-8167-2836.

† Bolsista, PIBIC/Fundação Araucária.  Universidade Tecnológica Federal do Paraná, Ponta Grossa, Paraná, Brasil. ✉ cristiellensilva@alunos.utfpr.edu.br. ☎ 0360460106414023.  0009-0002-5817-4233.

‡ Docente do Departamento de Engenharia Elétrica.  Universidade Tecnológica Federal do Paraná, Ponta Grossa, Paraná, Brasil. ✉ hugosiqueira@utfpr.edu.br. ☎ 6904980376005290.  0000-0002-1278-4602.

Séries temporais são conjuntos de dados que evoluem com o passar do tempo. Um exemplo de aplicação do poder computacional das RNAs é a sua utilização nesse tipo de dados, com o objetivo de prever valores futuros para variáveis selecionadas, utilizando uma base de dados com valores reais registrados. Tal aplicação ocorre devido a importância de se acompanhar a evolução de algumas variáveis ao longo do tempo, como o preço de moedas internacionais (SIQUEIRA *et al.*, 2018).

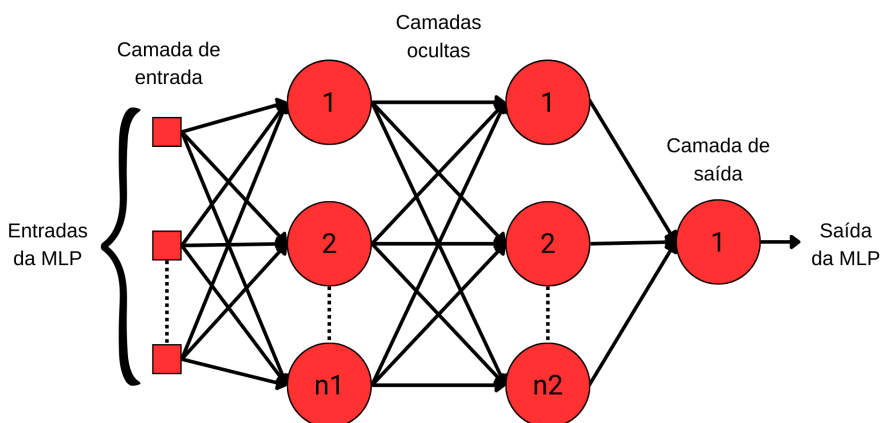
O presente relatório tem como objetivo utilizar um modelo de RNA para realizar a previsão de duas séries temporais distintas: a temperatura mínima diária da cidade de Melbourne, na Austrália (JASON BROWNEE, 2016), e o preço do dólar (YAHOO FINANCE, 2023).

## PERCEPTRON SIMPLES E DE MÚLTIPLAS CAMADAS

O modelo mais simples de uma RNA é o *Perceptron*. Ele recebe entradas ponderadas, as soma e aplica uma função de ativação para gerar uma saída. O *Perceptron* é capaz de aprender ajustando seus pesos de acordo com exemplos de treinamento, permitindo que a rede neural aprenda a realizar tarefas de classificação binária. Porém, existe uma grande limitação para esse modelo: sua incapacidade de lidar com problemas que não são linearmente separáveis. Ou seja, se os dados de entrada não forem separáveis por uma reta, seus resultados não serão satisfatórios (YANLING; BIMIN; ZHANRONG, 2002).

Para superar essa limitação, surge o Perceptron de Múltiplas Camadas (MLP, do inglês *Multilayer Perceptron*), uma extensão do conceito de *Perceptron*. A MLP incorpora camadas ocultas entre as camadas de entrada e de saída. Tais camadas adicionais permitem que a rede capture relações mais complexas e não lineares nos dados, tornando-o capaz de resolver problemas mais desafiadores. Cada neurônio na camada oculta processa informações de maneira iterativa, permitindo a criação de representações hierárquicas dos dados. Assim, a MLP supera as limitações do *Perceptron* simples, abrindo caminho para o processamento de informações mais sofisticadas e a resolução de tarefas mais diversificadas (HAYKIN, 2001). Por isso, para cumprir o objetivo de realizar as previsões de séries temporais, a MLP foi escolhida. A arquitetura desse modelo é representada na figura 1:

Figura 1 – Arquitetura da rede MLP.



Fonte: autoria própria (2023)

A MLP consiste em várias camadas de neurônios interconectadas de maneira *feed-forward* (propagação unidirecional da entrada para a saída), sendo elas divididas em três tipos diferentes: a camada de entrada, as camadas ocultas e a camada de saída.

A camada de entrada recebe os dados de entrada, que podem ser características, valores ou sinais. Cada neurônio nessa camada representa uma dimensão dos dados. Nas camadas ocultas cada neurônio recebe as saídas da camada anterior, calcula uma soma ponderada das entradas, aplica uma função de ativação não linear e envia para todos os neurônios da camada seguinte. A presença de múltiplas camadas ocultas permite que a MLP capture relações complexas e não lineares nos dados. A camada de saída gera as previsões ou classificações finais. Ela recebe as saídas da última camada oculta, realiza o mesmo processo de soma ponderada e ativação e produz as saídas finais da rede.

Além da arquitetura, a MLP tem outros fatores importantes para seu funcionamento: a função de ativação, os pesos e os *bias* (FLECK *et al.*, 2016).

A função de ativação introduz não-linearidade nas operações dos neurônios, permitindo que a rede modele relações complexas nos dados. Após calcular a soma ponderada das entradas, esta é passada através da função de ativação para determinar a saída do neurônio.

Os pesos e o *bias* determinam a influência de cada entrada na saída do neurônio. Estes são ajustados e otimizados durante o treinamento, de forma que o erro seja minimizado na previsão.

O algoritmo de treinamento funciona em basicamente duas etapas: a primeira, conhecida como fase de propagação direta (*forward*), as entradas são apresentadas e propagadas na rede, camada por camada, até mostrar uma saída. Durante essa etapa, os pesos são fixos e a saída obtida é comparada com a saída desejada, obtendo, assim, um valor de erro. Na segunda etapa, conhecida como fase de propagação reversa (*backward*), o erro é calculado e os pesos são ajustados. Essas duas fases acontecem a cada iteração do treinamento da rede. Esse algoritmo é conhecido como *backpropagation* (BATISTA, 2009).

Além do treinamento, também ocorre as etapas de validação e teste. A validação envolve a avaliação do desempenho do modelo em um conjunto de dados separado do conjunto de treinamento, permitindo ajustar hiperparâmetros para otimizar a capacidade de generalização do modelo, garantindo que ele funcione bem em situações do mundo real e não tenha sido superajustado aos dados de treinamento (GARDNER; DORLING, 1998). Já o teste consiste em avaliar o modelo em um conjunto de dados completamente novo, que não foi usado durante o treinamento ou a validação. O processo de teste e validação ajuda a assegurar que o modelo seja confiável e capaz de lidar com dados não vistos, contribuindo para tomadas de decisões mais acuradas.

## MÉTODOS

De início, as bases de dados foram tratadas de maneira que elas ficaram ideais para serem manipuladas. Após isso ser feito, cada base foi dividida da seguinte maneira: 70% dos dados para treinamento, 15% para validação e 15% para teste.

Em relação à função de ativação, foi utilizada a função tangente hiperbólica, uma função não-linear de forma sigmoideal que varia no intervalo (-1, 1). Por isso, na etapa de tratamento de dados, as bases foram normalizadas em valores entre -1 e 1. A fórmula matemática da tangente hiperbólica está na equação 1:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (1)$$

Para determinar a quantidade de neurônios da camada oculta, foi utilizada uma lógica *for*, rodando a rede utilizando de 5 até 100 neurônios, saltando de 5 em 5. A métrica utilizada para analisar o desempenho da rede foi o erro médio quadrado (MSE, do inglês *Mean Squared Error*), mostrada na equação 2. Assim, foi possível elencar o modelo mais eficiente dentre todos os testados.

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (2)$$

Também foi levada em consideração a quantidade de janelas deslizantes. Foram testadas redes com 3, 6 e 9 atrasos consecutivos, sendo a rede de 3 atrasos a que obteve resultados mais satisfatórios.

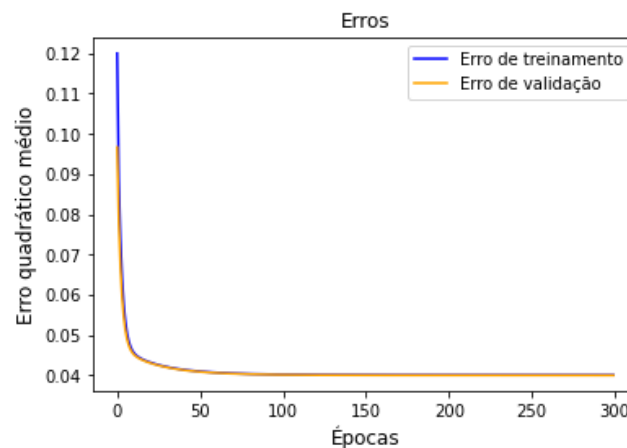
A quantidade de iterações foi definida como 300 para a previsão de temperatura e 100 para a previsão do preço do dólar, números escolhidos empiricamente.

## RESULTADOS

A fim de determinar a eficiência da rede, foram plotados dois gráficos para cada conjunto de dados: um de comparação entre os dados reais e os dados previstos, e outro com a evolução dos valores de erro de teste e de validação.

O gráfico de erros da série de temperatura e do preço do dólar estão representado nas figuras 2 e 3, respectivamente:

**Figura 2 – Evolução dos erros por época - dados de temperatura.**

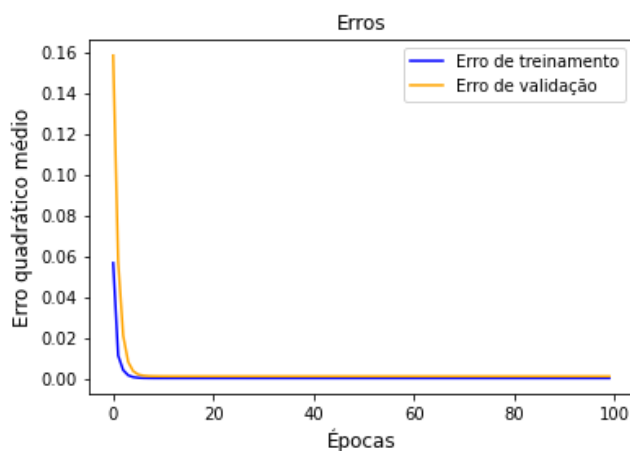


**Fonte: autoria própria (2023)**

É possível perceber que a quantidade de épocas escolhidas para cada uma das série foi suficiente para minimizar os erros de treinamento e validação e não foram demais a ponto de sobretreinar o modelo. O erro final, calculado utilizando o MSE, foi de 0,0331 para previsão de temperatura e 0,0012 para a previsão do preço do dólar.

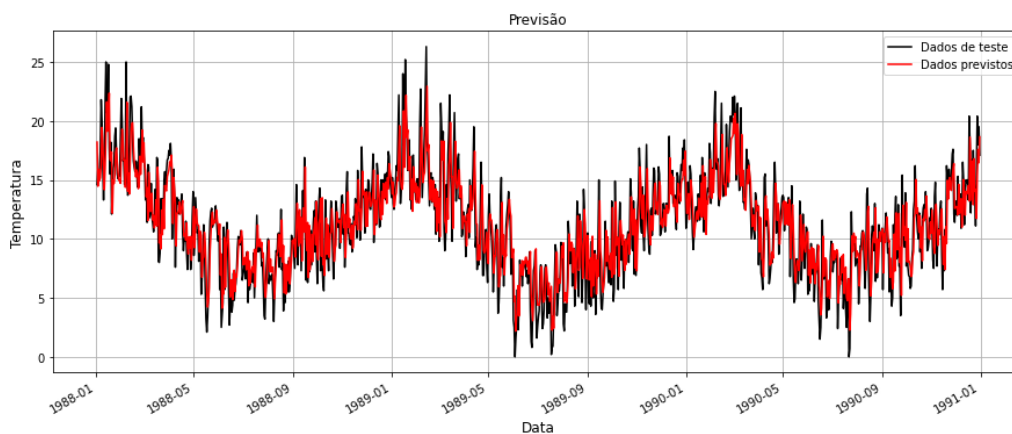
Os dados previstos foram plotados juntamente com os dados reais, a fim de ser possível uma comparação entre os dois conjuntos. As figuras 4 e 5 mostram essa comparação para as séries de temperatura e do preço do dólar, respectivamente. Note que a mesma conseguiu captar as relações estatísticas de maneira satisfatória.

**Figura 3 – Evolução dos erros por época - dados de preço do dólar.**



Fonte: autoria própria (2023)

**Figura 4 – Comparação entre dados reais e previstos - temperatura.**



Fonte: autoria própria (2023)

**Figura 5 – Comparação entre dados reais e previstos - preço do dólar.**



Fonte: autoria própria (2023)

## CONCLUSÃO

O presente relatório apresentou a aplicação de uma Rede Neural Artificial, um Perceptron de Múltiplas Camadas, para a previsão de dados de duas séries temporais: temperaturas mínimas diárias de uma cidade australiana, e o preço do dólar.

É possível observar que a rede tem uma alta capacidade de realizar previsões, visto que os erros foram de 0,0331 para as previsões de temperatura e 0,0012 para as previsões do preço do dólar.

Em relação a trabalhos futuros, é possível aplicar a citada rede neural artificial em outros problemas de previsão, com diferentes bases de dados.

## Agradecimentos

O presente trabalho foi realizado com apoio do CNPq, Conselho Nacional de Desenvolvimento Científico e Tecnológico — Brasil (Edital UTFPR/PROPPG n.o 04/2022 — PIBIT).

## Conflito de interesse

Não há conflito de interesse.

## REFERÊNCIAS

BATISTA, André Luiz França. Modelos de séries temporais e redes neurais artificiais na previsão de vazão. **Mestrado em engenharia de sistemas, Universidade Federal de Lavras, Lavras, Minas Gerais**, 2009.

FLECK, Leandro *et al.* Redes neurais artificiais: Principios básicos. **Revista Eletrônica Científica Inovação e Tecnologia**, v. 1, n. 13, p. 47–57, 2016.

GARDNER, Matt W; DORLING, SR. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. **Atmospheric environment**, Elsevier, v. 32, n. 14-15, p. 2627–2636, 1998.

HAYKIN, Simon. **Redes neurais: princípios e prática**. [S. l.]: Bookman Editora, 2001.

JASON BROWNLEE. **7 Time Series Datasets for Machine Learning**. [S. l.: s. n.], 2016. Disponível em: <https://machinelearningmastery.com/time-series-datasets-for-machine-learning/>. Acesso em: 13 de abril 2023.

SILVA, Ivan Nunes da; SPATTI, Danilo Hernane; FLAUZINO, Rogério Andrade. Redes neurais artificiais para engenharia e ciências aplicadas, 2010.

SIQUEIRA, Hugo *et al.* Performance analysis of unorganized machines in streamflow forecasting of Brazilian plants. **Applied Soft Computing**, Elsevier, v. 68, p. 494–506, 2018.

YAHOO FINANCE. **Yahoo Finance**. [S. l.: s. n.], 2023. Disponível em: <https://rb.gy/82apx>. Acesso em: 18 de junho 2023.

YANLING, Zhao; BIMIN, Deng; ZHANRONG, Wang. Analysis and study of perceptron to solve XOR problem. *In*: IEEE. THE 2nd International Workshop on Autonomous Decentralized System, 2002. [S. l.: s. n.], 2002. P. 168–173.