

Redes Neurais Multi Layer Perceptron Aplicadas a Previsão dos Preços de Ações

Luís F. S. Souza¹, Bárbara D. do A. Rodriguez², Cristiana A. Poffal³

Universidade Federal do Rio Grande, Rio Grande, RS

Uma Rede Neural Artificial (RNA) é composta por várias unidades de processamento (também conhecidas como neurônios artificiais), cujo funcionamento é bastante simples. Essas unidades, geralmente, são conectadas por canais de comunicação associados a um determinado peso. As unidades efetuam operações apenas sobre seus dados locais, entradas recebidas pelas suas conexões. O comportamento inteligente de uma RNA vem das interações entre as unidades de processamento da rede [1]. No aprendizado profundo, um Perceptron Multicamadas (MLP) é o nome de uma Rede Neural *feedforward* que consiste em neurônios totalmente conectados com funções de ativação não lineares, organizados em camadas, notáveis por serem capazes de distinguir dados que não são linearmente separáveis [3]. Essa Rede Neural *feedforward* distingue-se pela presença de uma ou mais camadas ocultas, cujos nós computacionais são chamados de neurônios ocultos ou unidades ocultas. A função desses neurônios é intervir entre a entrada externa e a saída da rede de uma maneira útil. Em um sentido bastante livre, a rede adquire uma perspectiva global apesar de sua conectividade local, devido ao conjunto extra de conexões sinápticas e da dimensão extra de interações neurais [2]. Os nós de fonte da camada de entrada da rede fornecem os respectivos elementos do padrão de ativação (vetor de entrada), que constituem os sinais de entrada aplicados aos neurônios (nós computacionais) na segunda camada (a primeira camada oculta). Os sinais de saída da segunda camada são utilizados como entradas para a terceira camada, e assim por diante para o resto da rede. Tipicamente, os neurônios em cada camada da rede têm como suas entradas apenas os sinais de saída da camada precedente. O conjunto de sinais de saída dos neurônios da camada de saída (final) da rede constitui a resposta global da rede para o padrão de ativação fornecido pelos nós de fonte da camada de entrada (primeira). Como exemplo de arquitetura de rede, uma rede *feedforward* com m nós de fonte, h_1 neurônios na primeira camada oculta, h_2 neurônios na segunda camada oculta e q neurônios na camada de saída é referida como uma rede $m - h_1 - h_2 - q$. As RNAs têm se tornado essenciais em diversas áreas devido à sua capacidade de resolver problemas complexos [5]. Elas podem, por exemplo, ser empregadas na previsão de séries temporais financeiras, uma vez que possuem grande poder computacional e capacidade de aprender para generalizar [4]. Na área econômica, o mercado de ações é um dos principais mecanismos para o desenvolvimento da economia, pois trata-se de um importante meio de captação de capitais. Desta forma, a tentativa de realizar a previsão do comportamento de preço neste mercado é de extrema importância [5]. Dentro deste contexto, este trabalho propõe utilizar o tipo de rede MLP para a previsão dos preços de ações de empresas na bolsa. Utilizando a linguagem de programação Python na *IDE* do Google Colab, foi implementado um modelo de RNA com arquitetura 100 – 50 – 1 com otimizador *ADAM* e função perda de *Mean Square Error*, 100 épocas de treinamento e taxa de aprendizado 0,001 como mostra a Figura 1. Essa rede foi treinada com os preços baseados nos dados da biblioteca *yfinance* (<https://pypi.org/project/yfinance/>) de fechamento da ação

¹luisfernandessaucedosouza@furg.br

²barbararodriguez@furg.br

³cristianaandrade@furg.br

WEGE3.SA do dia 01/01/2023 até 01/01/2024. A partir dessa rede treinada, tentou-se prever o preço dessa ação do dia 01/01/2024 até 01/06/2024. Os resultados são apresentados na Figura 2.

```

regressor = Sequential()
regressor.add(Dense(units = 100, activation = 'relu', input_dim = previsores.shape[1])) # Camada de entrada
regressor.add(Dropout(0.3)) # Dropout para regularização
regressor.add(Dense(units = 50, activation = 'relu')) # Camada oculta
regressor.add(Dropout(0.3))
regressor.add(Dense(units = 1, activation = 'linear')) # Camada de saída

regressor.compile(optimizer = 'adam', loss = "mean_squared_error", metrics = ['mean_absolute_error'])
regressor.fit(previsores, preco_real, epochs = 100, batch_size = 32)

```

Figura 1: Código para a estrutura de rede utilizada. Fonte: Dos Autores.

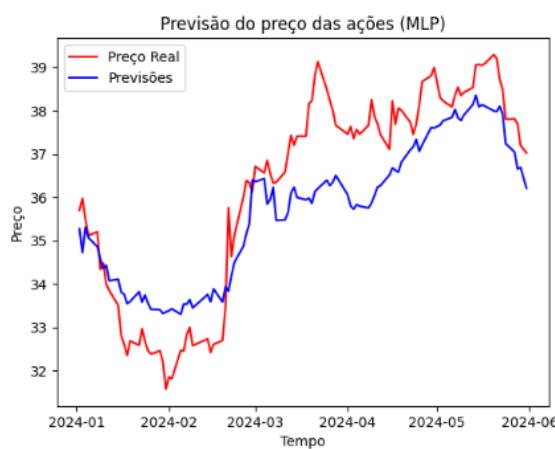


Figura 2: Preço real e previsão do preço de fechamento da ação WEG3.SA. Fonte: Dos Autores.

O seu *Mean Absolute Error (MAE)* foi em torno de 1,16, significando que em média por dia se tem um erro de R\$ 1,16 na previsão do preço da ação. O método, apesar de não apresentar grande precisão na previsão, demonstrou-se eficaz para analisar alguns comportamentos dos preços da ação, o que já auxiliaria tomadas de decisão.

Referências

- [1] A. P. de L. F. de C. Carvalho. **Redes neurais artificiais**. <https://sites.icmc.usp.br/andre/research/neural/>. Acessado em: 2 mar. 2025. 2020.
- [2] P. S. Churchland e T. J. Sejnowski. **The computational brain**. Cambridge: The MIT Press, 1992.
- [3] G. Cybenko. “Approximation by Superpositions of a Sigmoidal Function”. Em: **Mathematics of Control, Signals, and Systems** 2 (1989), pp. 303–314. DOI: 10.1007/BF02551274.
- [4] S. Haykin. **Redes Neurais - Princípios e práticas**. Porto Alegre: Bookman, 2007.
- [5] J. L. A. Moura. **Redes neurais: uma aplicação de visão computacional**. <https://repositorio.ufu.br/bitstream/123456789/44459/2/RedesNeuraisAplica.pdf>. Acessado em: 2 mar. 2025. 2024.